Guide de l'utilisateur pour la version 2

AWS Command Line Interface



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Command Line Interface: Guide de l'utilisateur pour la version 2

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques commerciales et la présentation commerciale d’Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d’une manière susceptible d’entraîner une confusion chez les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

..... xv
À propos de AWS CLI 1
- À propos de AWS CLI la version 2 1
- Maintenance et prise en charge des versions majeures du SDK 2
- À propos d'Amazon Web Services 2
- À propos des exemples 3
- Documentation et ressources supplémentaires 4
  - AWS CLI documentation et ressources 4
  - Autres AWS SDKs outils 5

Mise en route 6
- Prérequis 7
  - Création d'un compte administratif IAM ou IAM Identity Center 7
  - Étapes suivantes 8
- Installation/mise à jour 9
  - AWS CLI instructions d'installation et de mise à jour 9
  - Résolution des erreurs AWS CLI d'installation et de désinstallation 30
  - Étapes suivantes 30
- Versions passées 30
  - Résolution des erreurs AWS CLI d'installation et de désinstallation 46
  - Étapes suivantes 46
- Compiler et installer à partir du code source 47
  - Pourquoi créer à partir des sources ? 47
  - Étapes rapides 48
  - Étape 1 : Configuration de toutes les exigences 51
  - Étape 2 : Configuration de l'installation AWS CLI source 55
  - Étape 3 : Construire le AWS CLI 62
  - Étape 4 : Installation du AWS CLI 63
  - Étape 5 : vérification de l' AWS CLI installation 65
  - Exemples de flux de travail 65
  - Résolution des erreurs AWS CLI d'installation et de désinstallation 68
  - Étapes suivantes 68
- Amazon ECR Public/Docker 68
  - Prérequis 69
  - Choisir entre Amazon ECR Public et Docker Hub 69

Exécutez les images officielles 70
Remarques sur les interfaces et la rétrocompatibilité des images officielles 71
Utiliser des versions et des balises spécifiques 71
Mise à jour vers la dernière image officielle 72
Partage de fichiers hôtes, d'informations d'identification, de variables d'environnement et de configuration 73
Raccourcissez la commande docker run 79
Configuration 82
Rassemblez vos informations d'identification pour un accès programmatique 83
Configuration d'une nouvelle configuration et de nouvelles informations d'identification 85
Utilisation des fichiers de configuration et d'identification existants 93
Configurez le AWS CLI 94
Priorité de la configuration et des informations d'identification 94
Sujets supplémentaires dans cette section 95
Paramètres de configuration 96
Format des fichiers de configuration et d'identification 96
Où les paramètres de configuration sont-ils stockés ? 105
Utilisation de profils nommés 106
Définir et afficher les paramètres de configuration à l'aide de commandes 107
Exemples de commandes de définition d'une nouvelle configuration et d'informations d'identification 111
Paramètres de fichier `config` pris en charge 114
Variables d'environnement 135
Comment définir les variables d'environnement 136
AWS CLI variables d'environnement prises en charge 137
Options de ligne de commande 150
Comment utiliser les options de ligne de commande 150
AWS CLI options de ligne de commande globales prises en charge 151
Utilisations courantes des options de ligne de commande 156
Saisie automatique de commandes 156
Comment ça marche 157
Configuration de l'exécution des commandes sous Linux ou macOS 158
Configuration de l'exécution des commandes sous Windows 161
Nouvelle tentative 163
Modes de nouvelle tentative disponibles 163
Configuration d'un mode de nouvelle tentative 166

Afficher les journaux des tentatives de nouvelle tentative ... 167
Proxys HTTP ... 168
Utilisation des exemples ... 168
Authentification auprès d'un proxy ... 170
Utilisation d'un proxy sur les EC2 instances Amazon ... 170
Résolution des problèmes ... 171
Points de terminaison ... 171
Définir le point de terminaison pour une seule commande ... 172
Définir un point de terminaison global pour tous Services AWS ... 172
Paramétré pour utiliser les FIPs points de terminaison pour tous Services AWS ... 173
Conçu pour utiliser des points de terminaison à double pile pour tous Services AWS ... 175
Définissez des points de terminaison spécifiques au service ... 176
Points de terminaison basés sur des comptes ... 179
Configuration du point de terminaison et priorité des paramètres ... 181
Authentification et identifiants d'accès ... 183
Configuration et priorité des informations d'identification ... 185
Sujets supplémentaires dans cette section ... 186
Authentification IAM Identity Center ... 186
Prérequis ... 187
Configurez votre profil avec l'`aws configure sso`assistant ... 189
Configurez uniquement votre `sso-session` section avec `aws configure sso-session` l'assistant ... 193
Configuration manuelle à l'aide du `config` fichier ... 193
Connectez-vous à une session IAM Identity Center ... 196
Exécutez une commande avec votre profil IAM Identity Center ... 198
Déconnectez-vous de vos sessions IAM Identity Center ... 198
Résolution des problèmes ... 198
Ressources connexes ... 199
Concepts du centre d'identité IAM ... 199
Tutoriel : AWS IAM Identity Center et Amazon S3 ... 204
Identifiants à court terme ... 210
Rôles IAM ... 212
Prérequis ... 212
Vue d'ensemble de l'utilisation des rôles IAM ... 212
Configuration et utilisation d'un rôle ... 214
Utilisation de MFA ... 216

Rôles entre comptes et ID externe ... 218
Spécification d'un nom de session de rôle pour faciliter l'audit ... 218
Utilisation d'un rôle à l'aide d'une identité web ... 219
Suppression des informations d'identification mises en cache ... 221
Utilisateurs IAM ... 221
Étape 1 : créer votre utilisateur IAM ... 222
Étape 2 : Obtenez vos clés d'accès ... 222
Configurez le AWS CLI ... 223
EC2 Métadonnées Amazon ... 225
Prérequis ... 226
Configuration d'un profil pour les EC2 métadonnées Amazon ... 226
External Credentials ... 227
En utilisant le AWS CLI ... 230
Obtenir de l'aide ... 231
La commande d' AWS CLI aide intégrée ... 231
AWS CLI guide de référence ... 236
Documentation sur les API ... 236
Résolution des erreurs ... 237
Aide supplémentaire ... 237
Structure de commande ... 237
Structure de commande ... 238
Commandes d'attente ... 239
Spécifier les valeurs des paramètres ... 240
Types de paramètres courants ... 241
Citations avec chaînes ... 247
Paramètres du fichier ... 251
Générer un modèle de squelette de l'interface de ligne de commande ... 255
Syntaxe raccourcie ... 269
Invite automatique ... 272
Comment ça marche ... 272
Fonctionnalités d'invite automatique ... 273
Modes d'invite automatique ... 276
Configurer l'invite automatique ... 277
Sortie de commande de contrôle ... 277
Sortie sensible ... 277
Options de sortie côté serveur ou côté client ... 278

Résultat 279
Pagination 289
Sortie du filtre 295
Codes de retour 319
Magiciens 321
Comment ça marche 321
Alias 323
Prérequis 323
Étape 1 : Création du fichier d'alias 323
Étape 2 : Création d'un alias 325
Étape 3 : appeler un alias 328
Exemples de référentiels d'alias 330
Ressources 331
Résoudre les erreurs 332
Résolution des problèmes généraux à essayer en premier 332
Erreurs de commande non trouvées 341
La commande « `aws --version` » renvoie une version différente de celle que vous avez installée 344
La commande « `aws --version` » renvoie une version après la désinstallation du AWS CLI 345
Le a AWS CLI traité une commande avec un nom de paramètre incomplet 347
Erreurs d'accès refusé 348
Informations d'identification non valides et erreurs clés 349
La signature ne correspond pas aux erreurs 351
Erreurs de certificat SSL 352
Erreurs JSON non valides 354
Ressources supplémentaires 356
AWS CLI exemples 357
Exemples de commandes guidées 357
DynamoDB 358
Amazon EC2 362
S3 Glacier 382
IAM 389
Amazon S3 393
Amazon SNS 413
Exemples de commandes 416

ACM 422
API Gateway 434
API Gateway HTTP et WebSocket API 502
API de gestion d'API Gateway 556
App Mesh 558
App Runner 603
AWS AppConfig 639
Application Autoscaling 675
Application Discovery Service 692
Application Signals 699
AppRegistry 720
Athena 732
Auto Scaling 767
Plans de dimensionnement automatique 839
AWS Backup 847
AWS Batch 853
AWS Budgets 868
Amazon Chime 879
API de contrôle du cloud 926
AWS Cloud Map 932
AWS Cloud9 956
AWS CloudFormation 965
CloudFront 1016
Amazon CloudSearch 1124
CloudTrail 1125
CloudWatch 1143
CloudWatch Journaux 1179
CloudWatch Surveillance du réseau 1186
CloudWatch Moniteur d'accès à l'observabilité 1199
CloudWatch Administrateur de l'observabilité 1211
CloudWatch Synthetics 1217
CodeArtifact 1236
CodeBuild 1265
CodeCommit 1331
CodeDeploy 1407
CodeGuru Réviseur 1449

CodePipeline 1468
AWS CodeStar Notifications 1501
CodeConnections 1512
Amazon Cognito Identity 1521
Fournisseur d'identité Amazon Cognito 1527
Amazon Comprehend 1677
Amazon Comprehend Medical 1816
AWS Config 1853
Amazon Connect 1877
AWS Cost and Usage Report 1895
Service Cost Explorer 1898
Firehose 1906
Amazon Data Lifecycle Manager 1909
AWS Data Pipeline 1916
DataSync 1925
DAX 1930
Détective 1949
Device Farm 1962
AWS Direct Connect 1967
AWS Directory Service 2019
AWS Directory Service Données 2022
AWS DMS 2047
Amazon DocumentDB 2092
DynamoDB 2150
DynamoDB Streams 2248
Amazon EC2 2255
Amazon EC2 Instance Connect 2949
Amazon ECR 2950
Amazon ECR Public 2982
Amazon ECS 3012
Amazon EFS 3140
Amazon EKS 3149
Elastic Beanstalk 3249
Elastic Load Balancing - Version 1 3280
Elastic Load Balancing - Version 2 3309
Elastic Transcoder 3364

ElastiCache 3391
MediaStore 3499
Amazon EMR 3517
Amazon EMR on EKS 3567
EventBridge 3568
EventBridge Tuyaux 3575
Firewall Manager 3583
AWS FIS 3593
GameLift Serveurs Amazon 3612
Global Accelerator 3647
AWS Glue 3687
GuardDuty 3709
AWS Health 3729
HealthImaging 3737
HealthLake 3764
HealthOmics 3778
IAM 3849
IAM Access Analyzer 3997
Image builder 4035
Incident Manager 4077
Incident Manager Contacts 4100
Amazon Inspector 4125
AWS IoT 4175
AWS IoT Analytics 4364
Device Advisor 4392
AWS IoT data 4407
AWS IoT Events 4410
AWS IoT Events-Data 4436
AWS IoT Greengrass 4462
AWS IoT Greengrass V2 4551
AWS IoT Jobs SDK release 4577
AWS IoT SiteWise 4581
AWS IoT Things Graph 4632
AWS IoT Wireless 4661
Amazon IVS 4699
Chat Amazon IVS 4745

Diffusion en temps réel d'Amazon IVS 4759
Amazon Kendra 4811
Kinesis 4820
AWS KMS 4840
Lake Formation 4915
Lambda 4968
License Manager 5012
Lightsail 5025
Macie 5153
Amazon Managed Grafana 5158
MediaConnect 5160
MediaConvert 5176
MediaLive 5202
MediaPackage 5208
MediaPackage VOD 5223
MediaStore Plan de données 5236
MediaTailor 5242
Base de données de mémoire 5247
Amazon MSK 5285
Surveillance du flux réseau 5294
Network Manager 5311
OpenSearch Service 5350
Organizations 5364
AWS Outposts 5404
AWS Payment Cryptography 5408
AWS Payment Cryptography Plan de données 5429
Amazon Pinpoint 5439
Amazon Polly 5463
AWS Price List 5469
AWS Private CA 5474
AWS Proton 5482
QLDB 5495
Amazon RDS 5519
Services de données Amazon RDS 5723
Analyse des performances d’Amazon RDS 5727
Amazon Redshift 5742

Amazon Rekognition 5824
AWS RAM 5902
Explorateur de ressource 5927
Groupes de ressources 5950
Resource Groups Tagging API 5964
AWS RoboMaker 5968
Route 53 6005
Enregistrement de domaine Route 53 6020
Profils Route 53 6047
Route 53 Resolver 6059
Amazon S3 6105
Contrôle Amazon S3 6201
S3 Glacier 6218
Secrets Manager 6241
Security Hub 6271
Security Lake 6351
AWS Serverless Application Repository 6387
Service Catalog 6389
Service Quotas 6421
Amazon SES 6432
Bouclier 6445
Signer 6462
Snowball Edge 6472
Amazon SNS 6474
Amazon SQS 6497
Storage Gateway 6518
AWS STS 6521
Support 6531
Amazon SWF 6545
Systems Manager 6562
Amazon Textract 6741
Amazon Transcribe 6753
Amazon Translate 6798
Trusted Advisor 6799
Verified Permissions 6820
VPC Lattice 6845

AWS WAF Classic 6882
AWS WAF Classic Regional 6888
AWS WAFV2 6894
WorkDocs 6941
Amazon WorkMail 6974
Flux de WorkMail messages Amazon 7000
WorkSpaces 7002
X-Ray 7017
Exemples de scripts Bash 7035
AWS Batch 7036
AWS Cloud Map 7048
CloudFront 7071
DynamoDB 7079
Amazon EC2 7178
HealthImaging 7336
IAM 7345
AWS KMS 7410
Lightsail 7413
Amazon S3 7421
AWS STS 7452
Sécurité 7456
Protection des données 7457
Chiffrement des données 7458
Gestion de l’identité et des accès 7458
Public ciblé 7459
Authentification par des identités 7459
Gestion des accès à l’aide de politiques 7463
Comment Services AWS travailler avec IAM 7467
Résolution des problèmes AWS d'identité et d'accès 7467
Validation de la conformité 7469
Résilience 7470
Sécurité de l’infrastructure 7471
Application d'une version minimale de TLS 7472
Guide de migration 7473
Nouvelles fonctionnalités et modifications 7473
AWS CLI nouvelles fonctionnalités de la version 2 7474

Changements majeurs entre AWS CLI la version 1 et AWS CLI la version 2 7476
Instructions de migration 7484
Remplacement de la version 1 par la version 2 7484
ide-by-sideInstallation S 7485
Désinstaller 7486
Résolution des erreurs AWS CLI d'installation et de désinstallation 7490
Historique du document 7491

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.

Qu'est-ce que la AWS Command Line Interface ?

The AWS Command Line Interface (AWS CLI) est un outil open source qui vous permet d'interagir avec les AWS services à l'aide de commandes dans votre shell de ligne de commande. Avec une configuration minimale, vous pouvez AWS CLI commencer à exécuter des commandes qui implémentent des fonctionnalités équivalentes à celles fournies par le navigateur à AWS Management Console partir de l'invite de commande de votre programme de terminal :

- Shells Linux : utilisez des programmes shell courants tels que `bash`zsh, et `tcsh`pour exécuter des commandes sous Linux ou macOS.
- Ligne de commande Windows : sous Windows, exécutez des commandes à partir de l'invite de commande Windows ou dans PowerShell.
- À distance : exécutez des commandes sur des instances Amazon Elastic Compute Cloud (Amazon EC2) via un programme de terminal distant tel que PuTTY ou SSH, ou avec. AWS Systems Manager

Toutes les fonctions d' AWS administration, de gestion et d'accès de l'IaaS (infrastructure en tant que service) AWS Management Console sont disponibles dans l' AWS API et. AWS CLI AWS Les nouvelles fonctionnalités et services IaaS fournissent des AWS Management Console fonctionnalités complètes via l'API et la CLI au lancement ou dans les 180 jours suivant le lancement.

AWS CLI Il fournit un accès direct aux AWS services publics APIs . Vous pouvez explorer les fonctionnalités d'un service à l' AWS CLI aide de et développer des scripts shell pour gérer vos ressources. Outre les commandes de bas niveau équivalentes à une API, plusieurs AWS services proposent des personnalisations pour le. AWS CLI Ces personnalisations peuvent inclure des commandes de plus haut niveau qui facilitent l'utilisation d'un service à l'aide d'une API complexe.

À propos de AWS CLI la version 2

La AWS CLI version 2 est la version majeure la plus récente AWS CLI et prend en charge toutes les dernières fonctionnalités. Certaines fonctionnalités introduites dans la version 2 ne sont pas rétroportées vers la version 1 et vous devez effectuer une mise à niveau pour accéder à ces fonctionnalités. Certaines modifications « décisives » par rapport à la version 1 peuvent vous obliger à modifier vos scripts. Pour obtenir la liste des principales modifications apportées à la version 2, consultezGuide de migration pour la AWS CLI version 2.

La AWS CLI version 2 ne peut être installée qu'en tant que programme d'installation groupé. Bien que vous puissiez le trouver dans les gestionnaires de packages, il s'agit de packages non pris en charge et non officiels qui ne sont ni produits ni gérés par. AWS Nous vous recommandons d'installer le uniquement AWS CLI à partir des points AWS de distribution officiels, comme indiqué dans ce guide.

Pour installer la AWS CLI version 2, voirthe section called “Installation/mise à jour”.

Pour vérifier la version actuellement installée, utilisez la commande suivante :

```
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Pour l'historique des versions, consultez le journal des modifications de la AWS CLI version 2 sur GitHub.

Maintenance et prise en charge des versions majeures du SDK

Pour plus d'informations sur la maintenance et le support des versions majeures du SDK et de leurs dépendances sous-jacentes, consultez les informations suivantes dans le guide de référence AWS SDKs and Tools :

- AWS SDKs et politique de maintenance des outils
- AWS SDKs et matrice de support des versions d'outils

À propos d'Amazon Web Services

Amazon Web Services (AWS) est un ensemble de services d'infrastructure numérique que les développeurs peuvent exploiter lors du développement de leurs applications. Les services incluent l'informatique, le stockage, les bases de données et la synchronisation des applications (messagerie et mise en file d'attente). AWS utilise un modèle pay-as-you-go de service. Seuls les services que vous utilisez ou que vos applications utilisent vous sont facturés. De plus, pour la rendre AWS plus accessible en tant que plate-forme de prototypage et d'expérimentation, AWS propose un niveau d'utilisation gratuit. Dans cette offre, les services sont gratuits en-dessous d'un certain niveau d'utilisation. Pour plus d'informations sur AWS les coûts et le niveau gratuit, consultez la section Niveau AWS gratuit. Pour obtenir un AWS compte, ouvrez la page d'AWS accueil, puis choisissez Créer un AWS compte.

À propos des exemples présentés dans le guide de AWS CLI l'utilisateur

Les exemples AWS Command Line Interface (AWS CLI) présentés dans ce guide sont formatés selon les conventions suivantes :

- Invite — L'invite de commande utilise l'invite Linux et s'affiche sous la forme (`$ `). Pour les commandes spécifiques à Windows, `C:\>` est utilisé comme invite. N'incluez pas le symbole d'invite lorsque vous saisissez des commandes.
- Répertoire : lorsque des commandes doivent être saisies depuis un répertoire spécifique, le nom de répertoire s'affiche sous le symbole d'invite.
- Entrée utilisateur : le texte de commande que vous saisissez en ligne de commande est indiqué sous la forme **`user input`**.
- Texte remplaçable : le texte variable, y compris les noms des ressources que vous choisissez ou IDs générées par les AWS services que vous devez inclure dans les commandes, est formaté comme suit. *`replaceable text`* Dans les commandes sur plusieurs lignes ou les commandes dans lesquelles une saisie clavier spécifique est nécessaire, les commandes clavier peuvent également être indiquées sous la forme de texte remplaçable.
- Sortie — La sortie renvoyée par les AWS services est affichée sous la forme saisie par l'utilisateur et formatée comme `computer output` suit.

L'exemple de **`aws configure`** commande suivant illustre la saisie par l'utilisateur, le texte remplaçable et la sortie :

1. Entrez **`aws configure`** sur la ligne de commande, puis appuyez sur Entrée.
2. Il AWS CLI affiche des lignes de texte qui vous invitent à saisir des informations supplémentaires.
3. Saisissez chaque clé d'accès tour à tour, puis appuyez sur Entrée.
4. Entrez ensuite un nom de AWS région dans le format indiqué, appuyez sur Entrée, puis appuyez une dernière fois sur Entrée pour ignorer le réglage du format de sortie.
5. La commande Entrée de fin apparaît sous la forme d'un texte remplaçable car il n'y a aucune entrée utilisateur pour cette ligne.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: ENTER
```

L'exemple suivant illustre une commande simple avec sortie. Pour utiliser cet exemple, saisissez le texte intégral de la commande (texte en surbrillance après l'invite) et appuyez sur Entrée. Le nom du groupe de sécurité, *my-sg*, peut être remplacé par le nom du groupe de sécurité souhaité. Le document JSON, y compris les accolades, est produit en sortie. Si vous configurez votre interface CLI pour une sortie au format texte ou tableau, la sortie sera mise en forme différemment. JSON est le format de sortie par défaut.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group"
{
    "GroupId": "sg-903004f8"
}
```

Documentation et ressources supplémentaires pour AWS CLI

AWS CLI documentation et ressources

Outre ce guide de l'utilisateur, vous trouverez ci-dessous de précieuses ressources en ligne lorsque vous utilisez le AWS CLI.

- AWS CLI guide de référence version 2
- AWS CLI Référentiel d'exemples de code de script Bash. Exemples de scripts bash open source. Les exemples de scripts Bash sont hébergés dans le référentiel d'exemples de AWS code sur. GitHub
- AWS CLI GitHub référentiel. Vous pouvez afficher et bifurquer le code source pour AWS CLI le faire GitHub. Rejoignez la communauté d'utilisateurs GitHubpour faire part de vos commentaires, demander des fonctionnalités et soumettre vos propres contributions. Cela inclut l'affichage et la fourniture d'exemples de commandes pour la AWS CLI documentation.
- AWS CLI référentiel d'exemples d'alias Vous pouvez consulter et bifurquer des exemples d' AWS CLI alias sur GitHub.
- AWS CLI version 2 Changelog

Autres AWS SDKs outils

En fonction de votre cas d'utilisation, vous souhaiterez peut-être choisir l'un AWS SDKs des outils les mieux adaptés à vos besoins :

- AWS SDKs Guide de référence et d'outils
- AWS SDK pour C++
- AWS SDK pour Go
- AWS SDK pour Java
- AWS SDK pour JavaScript
- AWS SDK pour Kotlin
- AWS SDK pour .NET
- AWS SDK for Python (Boto)
- AWS SDK pour PHP
- Outils AWS pour PowerShell
- AWS SDK pour Ruby
- Kit AWS SDK pour Rust
- AWS SDK pour SAP ABAP
- Kit AWS SDK pour Swift
- AWS Amplify

Commencer à utiliser le AWS CLI

Ce chapitre décrit les étapes à suivre pour démarrer avec la version 2 de AWS Command Line Interface (AWS CLI) et fournit des liens vers les instructions pertinentes.

1. Remplissez tous les prérequis : pour accéder aux AWS services avec le AWS CLI, vous devez disposer au minimum d'informations d'identification AN Compte AWS et IAM. Pour renforcer la sécurité de votre AWS compte, nous vous recommandons de ne pas utiliser les informations d'identification de votre compte root. Vous devez créer un utilisateur ayant le moins de privilèges pour fournir des informations d'accès aux tâches que vous allez exécuter AWS.
2. Installez le ou accédez-y AWS CLI en utilisant l'une des méthodes suivantes :
   - (Recommandé)the section called “Installation/mise à jour”.
   - the section called “Versions passées”. L'installation d'une version spécifique est principalement utilisée si votre équipe aligne ses outils sur une version spécifique.
   - the section called “Compiler et installer à partir du code source”. La création AWS CLI à partir des GitHub sources est une méthode plus approfondie qui est principalement utilisée par les clients qui travaillent sur des plateformes que nous ne prenons pas directement en charge par nos installateurs prédéfinis.
   - the section called “Amazon ECR Public/Docker”.
   - Accédez à la AWS CLI version 2 dans la AWS console depuis votre navigateur en utilisant AWS CloudShell. Pour plus d’informations, consultez le Guide de l’utilisateur AWS CloudShell.
3. Une fois que vous y avez accès AWS CLI, configurez votre compte AWS CLI avec vos informations d'identification IAM pour la première utilisation.

> **Résolution des erreurs d'installation ou de configuration**
>
> Si vous rencontrez des problèmes après l'installation, la désinstallation ou la configuration du AWS CLI, consultez les étapes de résolution the section called “Résoudre les erreurs” des problèmes.

Rubriques

- Prérequis pour utiliser la AWS CLI version 2
- Installation ou mise à jour vers la dernière version du AWS CLI

- Installation des versions précédentes de la AWS CLI version 2
- Construire et installer le AWS CLI depuis le code source
- Exécution des images officielles Amazon ECR Public ou Docker pour AWS CLI
- Configuration du AWS CLI

Prérequis pour utiliser la AWS CLI version 2

Pour accéder aux AWS services avec le AWS CLI, vous avez besoin d'informations d'identification Compte AWS et IAM. Lors de l' AWS CLI exécution de AWS CLI commandes, ils doivent avoir accès à ces AWS informations d'identification. Pour renforcer la sécurité de votre AWS compte, nous vous recommandons de ne pas utiliser les informations d'identification de votre compte root. Vous devez créer un utilisateur ayant le moins de privilèges pour fournir des informations d'accès aux tâches que vous allez exécuter AWS.

Rubriques

- Création d'un compte administratif IAM ou IAM Identity Center
- Étapes suivantes

Création d'un compte administratif IAM ou IAM Identity Center

Avant de pouvoir configurer le AWS CLI, vous devez créer un compte IAM ou IAM Identity Center.

Afin de créer un utilisateur administrateur, choisissez l'une des options suivantes :

Choisissez un moyen de gérer votre administrateur	Pour	Par	Vous pouvez également
Dans IAM Identity Center	Utiliser des identifiants à court terme pour accéder à AWS.	Suivre les instructions de la section Mise en route dans le AWS IAM Identity Center Guide de l'utilisateur.	Configurez l'accès par programmation en configurant le AWS CLI à utiliser AWS IAM Identity Center dans le

Choisissez un moyen de gérer votre administrateur	Pour	Par	Vous pouvez également
(Recommandé)	Telles sont les meilleures pratiques en matière de sécurité. Pour plus d'informations sur les bonnes pratiques, veuillez consulter Security best practices in IAM (français non garanti) dans le Guide de l'utilisateur IAM.		guide de l'AWS Command Line Interface utilisateur.
Dans IAM (Non recommandé)	Utiliser des identifiants à long terme pour accéder à AWS.	Suivez les instructions de la section Créer un utilisateur IAM pour un accès d'urgence dans le guide de l'utilisateur IAM.	Configurez l'accès programmatique en gérant les clés d'accès pour les utilisateurs IAM dans le guide de l'utilisateur IAM.

Étapes suivantes

Après avoir créé des informations d'identification Compte AWS et IAM, pour les utiliser, AWS CLI vous pouvez effectuer l'une des opérations suivantes :

- Installez la dernière version de la AWS CLI version 2 sur votre ordinateur.
- Installez une version antérieure de la AWS CLI version 2 sur votre ordinateur.
- Accédez à la AWS CLI version 2 depuis votre ordinateur à l'aide d'une image Docker.

- Accédez à la AWS CLI version 2 dans la AWS console depuis votre navigateur à l'aide de AWS CloudShell. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudShell.

Installation ou mise à jour vers la dernière version du AWS CLI

Cette rubrique décrit comment installer ou mettre à jour la dernière version de AWS Command Line Interface (AWS CLI) sur les systèmes d'exploitation pris en charge. Pour plus d'informations sur les dernières versions de AWS CLI, consultez le journal des modifications de la AWS CLI version 2 sur GitHub.

Pour installer une version antérieure du AWS CLI, voirthe section called “Versions passées”. Pour les instructions de désinstallation, voirDésinstaller.

> **Important**
>
> AWS CLI les versions 1 et 2 utilisent le même nom de `aws` commande. Si vous avez déjà installé AWS CLI la version 1, consultezGuide de migration pour la AWS CLI version 2.

Rubriques

- AWS CLI instructions d'installation et de mise à jour
- Résolution des erreurs AWS CLI d'installation et de désinstallation
- Étapes suivantes

AWS CLI instructions d'installation et de mise à jour

Pour les instructions d'installation, développez la section correspondant à votre système d'exploitation.

Linux

Exigences d'installation et de mise à jour

- Vous devez être en mesure d'extraire ou de « décompresser » le package téléchargé. Si votre système d'exploitation ne dispose pas de la commande `unzip` intégrée, utilisez un équivalent.
- Les AWS CLI utilisations `glibc`, et`groff`. `less` Ceux-ci sont inclus par défaut dans la plupart des distributions majeures de Linux.

- Nous prenons en charge AWS CLI les versions 64 bits des distributions récentes de CentOS, Fedora, Ubuntu, Amazon Linux 1, Amazon Linux 2, Amazon Linux 2023 et Linux ARM.
- Dans la mesure où il AWS ne gère pas de référentiels tierssnap, nous ne pouvons garantir qu'ils contiennent la dernière version du AWS CLI.

Installez ou mettez à jour AWS CLI

> **Warning**
>
> S'il s'agit de votre première mise à jour sur Amazon Linux, pour installer la dernière version du AWS CLI, vous devez désinstaller la yum version préinstallée à l'aide de la commande suivante :
>
> ```
> $ sudo yum remove awscli
> ```
>
> Une fois l'yum AWS CLI installation supprimée, suivez les instructions d'installation de Linux ci-dessous.

Vous pouvez installer le AWS CLI en utilisant l'une des méthodes suivantes :

- L'installateur en ligne de commande est une bonne option pour le contrôle de version, car vous pouvez spécifier la version à installer. Cette option ne se met pas à jour automatiquement et vous devez télécharger un nouveau programme d'installation à chaque mise à jour pour remplacer la version précédente.
- Le **snap** package officiellement pris en charge est une bonne option pour toujours avoir la dernière version du package AWS CLI car les packages Snap sont automatiquement actualisés. Il n'existe aucun support intégré pour sélectionner des versions mineures de AWS CLI et n'est donc pas une méthode d'installation optimale si votre équipe a besoin d'épingler des versions.

Command line installer - Linux x86 (64-bit)

Pour mettre à jour votre installation actuelle de AWS CLI, téléchargez un nouveau programme d'installation à chaque mise à jour pour remplacer les versions précédentes. Suivez ces étapes depuis la ligne de commande pour installer AWS CLI le sous Linux.

Vous trouverez ci-dessous des étapes d'installation rapides dans un seul groupe de copier-coller qui fournit une installation de base. Pour obtenir des instructions guidées, reportez-vous aux étapes ci-dessous.

> **Note**
>
> (Facultatif) La commande suivante bloque le téléchargement et l'installe AWS CLI sans vérifier au préalable l'intégrité de votre téléchargement. Pour vérifier l'intégrité de votre téléchargement, suivez les instructions détaillées ci-dessous.

Pour installer le AWS CLI, exécutez les commandes suivantes.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Pour mettre à jour votre installation actuelle du AWS CLI, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande à l'aide des `--update` paramètres `--bin-dir``--install-dir`, et. Le bloc de commande suivant utilise un exemple de lien symbolique */usr/local/bin* et un exemple d'emplacement du programme d'installation */usr/local/aws-cli* pour installer le fichier AWS CLI localement pour l'utilisateur actuel.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --
update
```

Étapes d'installation guidées

1. Téléchargez le fichier d'installation en utilisant une des manières suivantes :

   - Utiliser la **`curl`** commande — L'`-o`option indique le nom de fichier dans lequel le package téléchargé est écrit. Les options de l'exemple de commande suivant permettent de copier le fichier téléchargé dans le répertoire courant avec le nom local `awscliv2.zip`

     ```
     $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
      "awscliv2.zip"
     ```

- Téléchargement depuis l'URL — Pour télécharger le programme d'installation avec votre navigateur, utilisez l'URL suivante : https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip

2. (Facultatif) Vérification de l'intégrité du fichier zip que vous avez téléchargé

   Si vous avez choisi de télécharger manuellement le package AWS CLI d'installation `.zip` en suivant les étapes ci-dessus, vous pouvez utiliser les étapes suivantes pour vérifier les signatures à l'aide de l'GnuPGoutil.

   Les `.zip` fichiers du package AWS CLI d'installation sont signés cryptographiquement à l'aide de signatures PGP. En cas de dommage ou d'altération des fichiers, cette vérification échoue et vous ne devez pas poursuivre l'installation.

   a. Téléchargez et installez la commande gpg à l'aide de votre gestionnaire de packages. Pour de plus amples informations sur GnuPG, veuillez consulter le site Web GnuPG.

   b. Pour créer le fichier de clé publique, créez un fichier texte et collez le texte suivant.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgusOJl7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4ciIx5OuLlnJZIxSzx
PqGl0mkxImLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LblHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJOz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtWO
lrFj6UwAsGukBTAOxC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutlIqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBFqglXmZ7l9ywG
EEUJYOlb2XrSuPWml39beWdKM8kzr1OjnlOm6+lpTRCBfo0wa9F8YZRhHPAkwKkX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbGlAYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
aGveYQUJDMpiLAAKCRCmMQrMRnJHXKBYD/9Ab0qQdGiO5hObchG8xh8Rpb4Mjyf6
0JrVo6m8GNjNj6BHkSc8fuTQJ/FaEhaQxj3pjZ3GXPrXjIIVChmICLlFuRXYzrXc
Pw0lniybypsZEVai5kO0tCNBCCFuMN9RsmmRG8mf7lC4FSTbUDmxG/QlYK+0IV/l
uJkzxWa+rySkdpm0JdqumjegNRgObdXHAQDWlubWQHWyZyIQ2B4U7AxqSpcdJp6I
S4Zds4wVLd1WE5pquYQ8vS2cNlDm4QNg8wTj58e3lKN47hXHMIb6CHxRnb947oJa
pg189LLPR5koh+EorNkA1wu5mAJtJvy5YMsppy2y/kIjp3lyY6AmPT1posgGk70Z
CmToEZ5rbd7ARExtlh76A0cabMDFlEHDIK8RNUOSRr7L64+KxOUegKBfQHb9dADY
qqiKqpCbKgvtWlds909Ms74JBgr2KwZCSY1HaOxnIr4CY43QRqAq5YHOay/mU+6w
hhmdF18vpyK0vfkvvGresWtSXbag7Hkt3XjaEw76BzxQH21EBDqU8WJVjHgU6ru+
```

```
DJTs+SxgJbaT3hb/vyjlw0lK+hFfhWKRwgOXH8vqducF95NRSUxtS4fpqxWVaw3Q
V2OWSjbne99A5EPEySzryFTKbMGwaTlAwMCwYevt4YT6eb7NmFhTx0Fis4TalUs+
j+c7Kg92pDx2uQ==
=OBAt
-----END PGP PUBLIC KEY BLOCK-----
```

Pour référence, les détails de la clé publique sont les suivants.

```
Key ID:          A6310ACC4672475C
Type:            RSA
Size:            4096/4096
Created:         2019-09-18
Expires:         2026-07-07
User ID:         AWS CLI Team <aws-cli@amazon.com>
Key fingerprint:  FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

c. Importez la clé AWS CLI publique à l'aide de la commande suivante, en la *public-key-file-name* remplaçant par le nom de fichier de la clé publique que vous avez créée.

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
 imported
gpg: Total number processed: 1
gpg:               imported: 1
```

d. Téléchargez le fichier de AWS CLI signature du package que vous avez téléchargé. Celui-ci a le même chemin d'accès et le même nom que le fichier `.zip` auquel il correspond, mais son extension est `.sig`. Dans les exemples suivants, nous l'enregistrons dans le répertoire courant sous la forme d'un fichier nommé `awscliv2.sig`.

Pour la dernière version de AWS CLI, utilisez le bloc de commande suivant :

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-
x86_64.zip.sig
```

e. Vérifiez la signature en transmettant les noms de fichiers `.sig` et `.zip` téléchargés sous forme de paramètres à la commande `gpg`.

```
$ gpg --verify awscliv2.sig awscliv2.zip
```

La sortie doit ressembler à ce qui suit :

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC
 4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
 owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

> **Important**
>
> L'avertissement mentionné dans la sortie est attendu et n'indique pas de problème. Cela se produit parce qu'il n'existe pas de chaîne de confiance entre votre clé PGP personnelle (si vous en avez une) et la clé AWS CLI PGP. Pour de plus amples informations, consultez Web of trust.

3. Décompressez le programme d'installation. Si votre distribution Linux n'a pas de commande `unzip` intégrée, utilisez un équivalent pour procéder à la décompression. L'exemple de commande suivant décompresse le package et crée un répertoire nommé `aws` dans le répertoire courant.

```
$ unzip awscliv2.zip
```

> **Note**
>
> Lors de la mise à jour à partir d'une version précédente, la `unzip` commande invite à remplacer les fichiers existants. Pour ignorer ces invites, par exemple pour l'automatisation des scripts, utilisez l'indicateur de `-u` mise à jour pour`unzip`. Cet indicateur met automatiquement à jour les fichiers existants et en crée de nouveaux selon les besoins.
>
> ```
> $ unzip -u awscliv2.zip
> ```

4. Exécutez le programme d'installation. La commande d'installation utilise un fichier nommé `install` dans le répertoire `aws` récemment décompressé. Par défaut, les fichiers sont tous installés dans `/usr/local/aws-cli` et un lien symbolique est créé dans `/usr/local/bin`. La commande inclut `sudo` pour accorder des autorisations d'écriture sur ces répertoires.

```
$ sudo ./aws/install
```

Vous pouvez effectuer l'installation sans `sudo` si vous spécifiez des répertoires sur lesquels vous disposez déjà d'autorisations en écriture. Utilisez les instructions suivantes pour la commande `install` de manière à spécifier l'emplacement d'installation :

- Assurez-vous que les chemins d'accès que vous fournissez aux paramètres `-i` et `-b` ne contiennent aucun nom de volume ou de répertoire incluant des espaces quelconques. Dans le cas contraire, l'installation échouera.
- `--install-dir` ou `-i` — Cette option indique le répertoire dans lequel copier tous les fichiers.

  La valeur par défaut est `/usr/local/aws-cli`.
- `--bin-dir` ou `-b` — Cette option indique que le `aws` programme principal du répertoire d'installation est symboliquement lié au fichier `aws` dans le chemin spécifié. Vous devez disposer des autorisations en écriture sur le répertoire spécifié. La création d'un lien symbolique vers un répertoire se trouvant déjà dans votre chemin d'accès vous dispense d'ajouter le répertoire d'installation à la variable $PATH de l'utilisateur.

  La valeur par défaut est `/usr/local/bin`.

```
$ ./aws/install -i /usr/local/aws-cli -b /usr/local/bin
```

> **Note**
>
> Pour mettre à jour votre installation actuelle du AWS CLI, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande avec le `--update` paramètre.
>
> ```
> $ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
> ```

Pour localiser le lien symbolique et le répertoire d'installation existants, procédez comme suit :

1. Utilisez la commande `which` pour rechercher votre lien symbolique. Cela vous indique le chemin à utiliser avec le paramètre `--bin-dir`.

```
$ which aws
/usr/local/bin/aws
```

2. Utilisez la commande `ls` pour rechercher le répertoire vers lequel votre lien symbolique pointe. Cela vous indique le chemin à utiliser avec le paramètre `--install-dir`.

```
$ ls -l /usr/local/bin/aws
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
```

5. Confirmez l'installation à l'aide de la commande suivante.

```
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Command line - Linux ARM

Pour mettre à jour votre installation actuelle de AWS CLI, téléchargez un nouveau programme d'installation à chaque mise à jour pour remplacer les versions précédentes. Suivez ces étapes depuis la ligne de commande pour installer AWS CLI le sous Linux.

Vous trouverez ci-dessous des étapes d'installation rapides dans un seul groupe de copier-coller qui fournit une installation de base. Pour obtenir des instructions guidées, reportez-vous aux étapes ci-dessous.

> **Note**
>
> (Facultatif) La commande suivante bloque le téléchargement et l'installe AWS CLI sans vérifier au préalable l'intégrité de votre téléchargement. Pour vérifier l'intégrité de votre téléchargement, suivez les instructions détaillées ci-dessous.

Pour installer le AWS CLI, exécutez les commandes suivantes.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Pour mettre à jour votre installation actuelle du AWS CLI, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande à l'aide des `--update` paramètres `--bin-dir``--install-dir`, et. Le bloc de commande suivant utilise un exemple de lien symbolique */usr/local/bin* et un exemple d'emplacement du programme d'installation de*/usr/local/aws-cli*.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --
update
```

Étapes d'installation guidées

1. Téléchargez le fichier d'installation en utilisant une des manières suivantes :

   - Utiliser la **`curl`** commande — L'`-o`option indique le nom de fichier dans lequel le package téléchargé est écrit. Les options de l'exemple de commande suivant permettent de copier le fichier téléchargé dans le répertoire courant avec le nom local `awscliv2.zip`

     ```
     $ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o
      "awscliv2.zip"
     ```

   - Téléchargement depuis l'URL — Pour télécharger le programme d'installation avec votre navigateur, utilisez l'URL suivante : https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip
2. (Facultatif) Vérification de l'intégrité du fichier zip que vous avez téléchargé

Si vous avez choisi de télécharger manuellement le package AWS CLI d'installation `.zip` en suivant les étapes ci-dessus, vous pouvez utiliser les étapes suivantes pour vérifier les signatures à l'aide de l'GnuPGoutil.

Les `.zip` fichiers du package AWS CLI d'installation sont signés cryptographiquement à l'aide de signatures PGP. En cas de dommage ou d'altération des fichiers, cette vérification échoue et vous ne devez pas poursuivre l'installation.

a. Téléchargez et installez la commande `gpg` à l'aide de votre gestionnaire de packages. Pour de plus amples informations sur GnuPG, veuillez consulter le site Web GnuPG.

b. Pour créer le fichier de clé publique, créez un fichier texte et collez le texte suivant.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgusOJl7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4ciIx5OuLlnJZIxSzx
PqGl0mkxImLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LblHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJOz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtWO
lrFj6UwAsGukBTAOxC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutlIqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBFqglXmZ7l9ywG
EEUJYOlb2XrSuPWml39beWdKM8kzr1OjnlOm6+lpTRCBfo0wa9F8YZRhHPAkwKkX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbGlAYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
aGveYQUJDMpiLAAKCRCmMQrMRnJHXKBYD/9Ab0qQdGiO5hObchG8xh8Rpb4Mjyf6
0JrVo6m8GNjNj6BHkSc8fuTQJ/FaEhaQxj3pjZ3GXPrXjIIVChmICLlFuRXYzrXc
Pw0lniybypsZEVai5kO0tCNBCCFuMN9RsmmRG8mf7lC4FSTbUDmxG/QlYK+0IV/l
uJkzxWa+rySkdpm0JdqumjegNRgObdXHAQDWlubWQHWyZyIQ2B4U7AxqSpcdJp6I
S4Zds4wVLd1WE5pquYQ8vS2cNlDm4QNg8wTj58e3lKN47hXHMIb6CHxRnb947oJa
pg189LLPR5koh+EorNkA1wu5mAJtJvy5YMsppy2y/kIjp3lyY6AmPT1posgGk70Z
CmToEZ5rbd7AREXtlh76A0cabMDFlEHDIK8RNUOSRr7L64+KxOUegKBfQHb9dADY
qqiKqpCbKgvtWlds909Ms74JBgr2KwZCSY1HaOxnIr4CY43QRqAq5YHOay/mU+6w
hhmdF18vpyK0vfkvvGresWtSXbag7Hkt3XjaEw76BzxQH21EBDqU8WJVjHgU6ru+
DJTs+SxgJbaT3hb/vyjlw0lK+hFfhWKRwgOXH8vqducF95NRSUxtS4fpqxWVaw3Q
V2OWSjbne99A5EPEySzryFTKbMGwaTlAwMCwYevt4YT6eb7NmFhTx0Fis4TalUs+
j+c7Kg92pDx2uQ==
=OBAt
-----END PGP PUBLIC KEY BLOCK-----
```

Pour référence, les détails de la clé publique sont les suivants.

```
Key ID:           A6310ACC4672475C
Type:             RSA
Size:             4096/4096
Created:          2019-09-18
Expires:          2026-07-07
User ID:          AWS CLI Team <aws-cli@amazon.com>
Key fingerprint:  FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

c. Importez la clé AWS CLI publique à l'aide de la commande suivante, en la *public-key-file-name* remplaçant par le nom de fichier de la clé publique que vous avez créée.

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
 imported
gpg: Total number processed: 1
gpg:               imported: 1
```

d. Téléchargez le fichier de AWS CLI signature du package que vous avez téléchargé. Celui-ci a le même chemin d'accès et le même nom que le fichier `.zip` auquel il correspond, mais son extension est `.sig`. Dans les exemples suivants, nous l'enregistrons dans le répertoire courant sous la forme d'un fichier nommé `awscliv2.sig`.

Pour la dernière version de AWS CLI, utilisez le bloc de commande suivant :

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-
aarch64.zip.sig
```

e. Vérifiez la signature en transmettant les noms de fichiers `.sig` et `.zip` téléchargés sous forme de paramètres à la commande gpg.

```
$ gpg --verify awscliv2.sig awscliv2.zip
```

La sortie doit ressembler à ce qui suit :

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
```

```
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC
 4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
 owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

> **Important**
>
> L'avertissement mentionné dans la sortie est attendu et n'indique pas de problème. Cela se produit parce qu'il n'existe pas de chaîne de confiance entre votre clé PGP personnelle (si vous en avez une) et la clé AWS CLI PGP. Pour de plus amples informations, consultez Web of trust.

3. Décompressez le programme d'installation. Si votre distribution Linux n'a pas de commande `unzip` intégrée, utilisez un équivalent pour procéder à la décompression. L'exemple de commande suivant décompresse le package et crée un répertoire nommé `aws` dans le répertoire courant.

```
$ unzip awscliv2.zip
```

> **Note**
>
> Lors de la mise à jour à partir d'une version précédente, la `unzip` commande invite à remplacer les fichiers existants. Pour ignorer ces invites, par exemple pour l'automatisation des scripts, utilisez l'indicateur de `-u` mise à jour pour`unzip`. Cet indicateur met automatiquement à jour les fichiers existants et en crée de nouveaux selon les besoins.
>
> ```
> $ unzip -u awscliv2.zip
> ```

4. Exécutez le programme d'installation. La commande d'installation utilise un fichier nommé `install` dans le répertoire `aws` récemment décompressé. Par défaut, les fichiers sont tous installés dans `/usr/local/aws-cli` et un lien symbolique est créé dans `/usr/local/bin`. La commande inclut `sudo` pour accorder des autorisations d'écriture sur ces répertoires.

```
$ sudo ./aws/install
```

Vous pouvez effectuer l'installation sans `sudo` si vous spécifiez des répertoires sur lesquels vous disposez déjà d'autorisations en écriture. Utilisez les instructions suivantes pour la commande `install` de manière à spécifier l'emplacement d'installation :

- Assurez-vous que les chemins d'accès que vous fournissez aux paramètres `-i` et `-b` ne contiennent aucun nom de volume ou de répertoire incluant des espaces quelconques. Dans le cas contraire, l'installation échouera.
- `--install-dir`ou `-i` — Cette option indique le répertoire dans lequel copier tous les fichiers.

  La valeur par défaut est `/usr/local/aws-cli`.
- `--bin-dir`ou `-b` — Cette option indique que le `aws` programme principal du répertoire d'installation est symboliquement lié au fichier `aws` dans le chemin spécifié. Vous devez disposer des autorisations en écriture sur le répertoire spécifié. La création d'un lien symbolique vers un répertoire se trouvant déjà dans votre chemin d'accès vous dispense d'ajouter le répertoire d'installation à la variable $PATH de l'utilisateur.

  La valeur par défaut est `/usr/local/bin`.

```
$ ./aws/install -i /usr/local/aws-cli -b /usr/local/bin
```

> **Note**
>
> Pour mettre à jour votre installation actuelle du AWS CLI, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande avec le `--update` paramètre.
>
> ```
> $ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/
> aws-cli --update
> ```
>
> Pour localiser le lien symbolique et le répertoire d'installation existants, procédez comme suit :

1. Utilisez la commande `which` pour rechercher votre lien symbolique. Cela vous indique le chemin à utiliser avec le paramètre `--bin-dir`.

```
$ which aws
/usr/local/bin/aws
```

2. Utilisez la commande `ls` pour rechercher le répertoire vers lequel votre lien symbolique pointe. Cela vous indique le chemin à utiliser avec le paramètre `--install-dir`.

```
$ ls -l /usr/local/bin/aws
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
```

5. Confirmez l'installation à l'aide de la commande suivante.

```
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Snap package

Nous fournissons une version officielle AWS prise en charge AWS CLI dusnap. Si vous souhaitez toujours avoir la dernière version AWS CLI installée sur votre système, un package Snap vous le fournit car il se met à jour automatiquement. Il n'existe pas de support intégré pour sélectionner des versions mineures de AWS CLI . Il ne s'agit donc pas d'une méthode d'installation optimale si votre équipe a besoin d'épingler des versions. Si vous souhaitez installer une version mineure spécifique du AWS CLI, nous vous conseillons d'utiliser le programme d'installation en ligne de commande.

1. Si votre plate-forme Linux n'est pas déjà `snap` installée, installez-la `snap` sur votre plate-forme.

a. Pour plus d'informations sur l'installation`snap`, consultez la section Installation du daemon dans la documentation de Snap.

b. Il se peut que vous deviez redémarrer votre système pour que vos `PATH` variables soient correctement mises à jour. Si vous rencontrez des problèmes d'installation, suivez les étapes décrites dans la section Résoudre les problèmes courants de la documentation Snap.

c. Pour vérifier qu'`snap`il est correctement installé, exécutez la commande suivante.

```
$ snap version
```

2. Exécutez la `snap install` commande suivante pour AWS CLI.

```
$ snap install aws-cli --classic
```

En fonction de vos autorisations, vous devrez peut-être ajouter des éléments `sudo` à la commande.

```
$ sudo snap install aws-cli --classic
```

> **Note**
>
> Pour consulter le référentiel de snapshots pour le AWS CLI, y compris `snap` les instructions supplémentaires, consultez la `aws-cli`page du site Web de Canonical Snapcraft.

3. Vérifiez qu'ils AWS CLI sont correctement installés.

```
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64
```

Si vous obtenez une erreur, consultez Résolution des erreurs liées au AWS CLI.

macOS

Exigences d'installation et de mise à jour

- Nous les prenons en charge AWS CLI sur les versions 11 et ultérieures de macOS. Pour plus d'informations, consultez les mises à jour de la politique de support de macOS pour la AWS CLI version v2 sur le blog AWS consacré aux outils de développement.
- Comme elle AWS ne gère pas de référentiels tiers, nous ne pouvons pas garantir qu'ils contiennent la dernière version du AWS CLI.

Matrice de prise en charge des versions de macOS

AWS CLI version	Version de macOS prise en charge
2.21.0 — actuel	11 ans et plus
2,17,0 —2,20,0	10,15 et plus
2,0.0 — 2,16.12	10.14 et versions antérieures

Installez ou mettez à jour AWS CLI

Si vous effectuez une mise à jour vers la dernière version, utilisez la même méthode d'installation que celle utilisée pour la version actuelle. Vous pouvez l'installer AWS CLI sur macOS de la manière suivante.

GUI installer

Les étapes suivantes indiquent comment installer la dernière version du à l'aide AWS CLI de l'interface utilisateur standard de macOS et de votre navigateur.

1. Dans votre navigateur, téléchargez le `pkg` fichier macOS : https://awscli.amazonaws.com/AWSCLIV2.pkg
2. Lancez le fichier téléchargé et suivez les instructions qui s'affichent à l'écran. Vous pouvez choisir d'installer AWS CLI le comme suit :
   - Pour tous les utilisateurs de l'ordinateur (nécessite **sudo**)

  - Vous pouvez installer dans n'importe quel dossier ou choisir le dossier par défaut recommandé, à savoir `/usr/local/aws-cli`.
  - Le programme d'installation crée automatiquement un lien symbolique à l'adresse `/usr/local/bin/aws` qui fait le lien avec le programme principal du dossier d'installation que vous avez choisi.
- Pour le seul utilisateur actuel (ne nécessite pas **sudo**)
  - Vous pouvez installer dans n'importe quel dossier sur lequel vous disposez d'une autorisation d'écriture.
  - En raison des autorisations utilisateur standard, une fois le programme d'installation terminé, vous devez créer manuellement un fichier de lien symbolique dans votre variable $PATH qui pointe vers les programmes `aws` et `aws_completer` en entrant les commandes suivantes à l'invite de commandes. L'emplacement par défaut d'un lien symbolique est `/usr/local/bin/` le suivant :

    ```
    $ ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
    $ ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/aws_completer
    ```

    Si vous ne disposez pas d'autorisations d'écriture sur le dossier, vous devrez peut-être l'utiliser `sudo` dans votre commande. L'exemple suivant utilise `sudo` l'emplacement par défaut d'un lien symbolique dans `/usr/local/bin/` :

    ```
    $ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
    $ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
    aws_completer
    ```

> **Note**
>
> Vous pouvez afficher les journaux de débogage pour l'installation en appuyant sur Cmd+L n'importe où dans le programme d'installation. Un volet du journal s'ouvre alors qui vous permet de filtrer et d'enregistrer le journal. Le fichier journal est également automatiquement enregistré dans `/var/log/install.log`.

3. Pour vérifier que le shell peut rechercher et exécuter la commande `aws` dans votre variable $PATH, utilisez les commandes suivantes.

    ```
    $ which aws
    ```

```
/usr/local/bin/aws
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Darwin/23.3.0
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Command line installer - All users

Si vous disposez des autorisations `sudo`, vous pouvez installer l' AWS CLI pour tous les utilisateurs sur l'ordinateur. Nous fournissons les étapes dans un groupe facile à copier et coller. Consultez les descriptions de chaque ligne dans les étapes suivantes.

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
$ sudo installer -pkg AWSCLIV2.pkg -target /
```

Instructions d'installation guidées

1. Vous pouvez télécharger le fichier à l'aide de la commande `curl`. L'option `-o` spécifie le nom de fichier dans lequel le package téléchargé est écrit. Dans cet exemple, le fichier est écrit dans `AWSCLIV2.pkg` du dossier actuel.

   ```
   $ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
   ```

2. Exécutez le programme `installer` macOS standard, en spécifiant le fichier `.pkg` téléchargé comme source. Utilisez le paramètre `-pkg` pour spécifier le nom du package à installer et le paramètre `-target /` pour spécifier l'unité sur laquelle le package doit être installé. Les fichiers sont installés dans `/usr/local/aws-cli`, et un lien symbolique est automatiquement créé dans `/usr/local/bin`. Vous devez inclure `sudo` dans la commande pour accorder les autorisations d'écriture à ces dossiers.

   ```
   $ sudo installer -pkg ./AWSCLIV2.pkg -target /
   ```

   Une fois l'installation terminée, les journaux de débogage sont écrits dans `/var/log/install.log`.

3. Pour vérifier que le shell peut rechercher et exécuter la commande `aws` dans votre variable `$PATH`, utilisez les commandes suivantes.

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Darwin/23.3.0
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Command line - Current user

1. Pour spécifier le dossier dans lequel AWS CLI il est installé, vous devez créer un fichier XML portant n'importe quel nom de fichier. Ce fichier est un fichier au format XML qui ressemble à l'exemple suivant. Laissez toutes les valeurs comme indiqué, sauf que vous devez remplacer le chemin */Users/myusername* de la ligne 9 par le chemin du dossier dans lequel vous souhaitez l' AWS CLI installer. Le dossier doit déjà exister, sinon la commande échoue. L'exemple XML suivant, nommé*choices.xml*, indique le programme d'installation à installer AWS CLI dans le dossier`/Users/myusername`, où il crée un dossier nomméaws-cli.

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
   PropertyList-1.0.dtd">
   <plist version="1.0">
     <array>
       <dict>
         <key>choiceAttribute</key>
         <string>customLocation</string>
         <key>attributeSetting</key>
         <string>/Users/myusername</string>
         <key>choiceIdentifier</key>
         <string>default</string>
       </dict>
     </array>
   </plist>
   ```

2. Téléchargez le programme d'installation `pkg` à l'aide de la commande `curl`. L'option `-o` spécifie le nom de fichier dans lequel le package téléchargé est écrit. Dans cet exemple, le fichier est écrit dans `AWSCLIV2.pkg` du dossier actuel.

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
```

3. Exécutez le programme `installer` macOS standard avec les options suivantes :

   - Spécifiez le nom du package à installer à l'aide du paramètre `-pkg`.
   - Spécifiez l'installation pour un utilisateur actuel uniquement en définissant le `-target` paramètre sur`CurrentUserHomeDirectory`.
   - Spécifiez le chemin d'accès (relatif au dossier actuel) et le nom du fichier XML que vous avez créé dans le paramètre `-applyChoiceChangesXML`.

   L'exemple suivant installe le AWS CLI dans le dossier. `/Users/myusername/aws-cli`

   ```
   $ installer -pkg AWSCLIV2.pkg \
               -target CurrentUserHomeDirectory \
               -applyChoiceChangesXML choices.xml
   ```

4. Les autorisations utilisateur standard n'autorisant généralement pas l'écriture dans les dossiers de votre variable `$PATH`, le programme d'installation de ce mode n'essaie pas d'ajouter les liens symboliques aux programmes `aws` et `aws_completer`. AWS CLI Pour que le programme s'exécute correctement, vous devez créer manuellement les liens symboliques une fois le programme d'installation terminé. Si votre variable `$PATH` inclut un dossier dans lequel vous pouvez écrire et que vous spécifiez ce dossier comme chemin d'accès de la cible, vous pouvez exécuter la commande suivante sans `sudo`. Si vous ne disposez pas d'un dossier accessible en écriture dans votre variable `$PATH`, vous devez utiliser `sudo` pour les autorisations d'écriture dans le dossier cible spécifié. L'emplacement par défaut d'un lien symbolique est `/usr/local/bin/`. `folder/installed`Remplacez-le par le chemin d'accès à votre AWS CLI installation.

   ```
   $ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
   $ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
   aws_completer
   ```

   Une fois l'installation terminée, les journaux de débogage sont écrits dans `/var/log/install.log`.

5. Pour vérifier que le shell peut rechercher et exécuter la commande `aws` dans votre variable `$PATH`, utilisez les commandes suivantes.

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Darwin/23.3.0
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Windows

Exigences d'installation et de mise à jour

- Nous prenons en charge les versions AWS CLI de Windows 64 bits prises en charge par Microsoft.
- Droits d'administrateur pour installer le logiciel

Installez ou mettez à jour AWS CLI

Pour mettre à jour votre installation actuelle AWS CLI sous Windows, téléchargez un nouveau programme d'installation à chaque mise à jour pour remplacer les versions précédentes. AWS CLI est régulièrement mis à jour. Pour savoir quand la dernière version a été publiée, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

1. Téléchargez et exécutez le programme d'installation AWS CLI MSI pour Windows (64 bits) :

   https://awscli.amazonaws.com/AWSCLIV2.msi

   Vous pouvez également exécuter la `msiexec` commande pour exécuter le programme d'installation MSI.

   ```
   C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
   ```

   Pour les différents paramètres pouvant être utilisés avec`msiexec`, consultez msiexec sur le site Web de Microsoft Docs. Par exemple, vous pouvez utiliser le `/qn` drapeau pour une installation silencieuse.

   ```
   C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn
   ```

2. Pour confirmer l'installation, ouvrez le menu Démarrer, recherchez `cmd` pour ouvrir une fenêtre d'invite de commandes et, à l'invite de commandes, utilisez la commande `aws --version`.

```
C:\> aws --version
aws-cli/2.27.41 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

   Si Windows ne trouve pas le programme, vous devrez peut-être fermer puis rouvrir la fenêtre d'invite de commande pour actualiser le chemin, ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Résolution des erreurs AWS CLI d'installation et de désinstallation

Si vous rencontrez des problèmes après l'installation ou la désinstallation du AWS CLI, consultez les étapes de résolution the section called “Résoudre les erreurs” des problèmes. Pour connaître les étapes de résolution des problèmes les plus pertinentesthe section called “Erreurs de commande non trouvées”, reportez-vous aux the section called “La commande « `aws --version` » renvoie une version différente de celle que vous avez installée” sections, etthe section called “La commande « `aws --version` » renvoie une version après la désinstallation du AWS CLI”.

Étapes suivantes

Une fois l'installation réussie AWS CLI, vous pouvez supprimer en toute sécurité les fichiers d'installation téléchargés. Après avoir terminé les étapes the section called “Prérequis” et installé le AWS CLI, vous devez effectuer unthe section called “Configuration”.

Installation des versions précédentes de la AWS CLI version 2

Cette rubrique décrit comment installer les versions précédentes de la AWS Command Line Interface version 2 (AWS CLI) sur les systèmes d'exploitation pris en charge. Pour plus d'informations sur les AWS CLI versions 2, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

AWS CLI instructions d'installation de la version 2 :

Linux

Exigences d'installation

- Vous savez quelle version de la AWS CLI version 2 vous souhaitez installer. Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

- Vous devez être en mesure d'extraire ou de « décompresser » le package téléchargé. Si votre système d'exploitation ne dispose pas de la commande `unzip` intégrée, utilisez un équivalent.
- La AWS CLI version 2 utilise `glibcgroff`, et`less`. Ceux-ci sont inclus par défaut dans la plupart des distributions majeures de Linux.
- Nous prenons en charge la AWS CLI version 2 sur les versions 64 bits des distributions récentes de CentOS, Fedora, Ubuntu, Amazon Linux 1, Amazon Linux 2 et Linux ARM.
- Comme elle AWS ne gère pas de référentiels tiers, nous ne pouvons pas garantir qu'ils contiennent la dernière version du AWS CLI.

Instructions d'installation

Suivez ces étapes depuis la ligne de commande pour installer AWS CLI le sous Linux.

Nous fournissons les étapes dans un groupe facile à copier et coller selon que vous utilisez Linux 64 bits ou Linux ARM. Consultez les descriptions de chaque ligne dans les étapes suivantes.

Linux x86 (64-bit)

> **Note**
>
> (Facultatif) La commande suivante bloque le téléchargement et l'installe AWS CLI sans vérifier au préalable l'intégrité de votre téléchargement. Pour vérifier l'intégrité de votre téléchargement, suivez les instructions détaillées ci-dessous.

Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur GitHub.

Pour installer le AWS CLI, exécutez les commandes suivantes.

Pour spécifier une version, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *2.0.30* entraînerait la commande suivante : `awscli-exe-linux-x86_64-2.0.30.zip`

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o
 "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Pour mettre à jour votre installation actuelle du AWS CLI, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande à l'aide des `--update` paramètres `--bin-dir--install-dir`, et. Le bloc de commande suivant utilise un exemple de lien symbolique */usr/local/bin* et un exemple d'emplacement du programme d'installation de*/usr/local/aws-cli*.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o
 "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --
update
```

Linux ARM

> **Note**
>
> (Facultatif) La commande suivante bloque le téléchargement et l'installe AWS CLI sans vérifier au préalable l'intégrité de votre téléchargement. Pour vérifier l'intégrité de votre téléchargement, suivez les instructions détaillées ci-dessous.

Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

Pour installer le AWS CLI, exécutez les commandes suivantes.

Pour spécifier une version, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *2.0.30* entraînerait la commande suivante : `awscli-exe-linux-aarch64-2.0.30.zip`

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip" -o
 "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Pour mettre à jour votre installation actuelle du AWS CLI, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande à l'aide des `--update` paramètres `--bin-dir--install-dir`, et. Le bloc de commande suivant utilise un exemple de lien symbolique */usr/local/bin* et un exemple d'emplacement du programme d'installation de*/usr/local/aws-cli*.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip" -o
 "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --
update
```

1. Téléchargez le fichier d'installation en utilisant une des manières suivantes :

   Linux x86 (64-bit)

   - Utiliser la **`curl`** commande — L'`-o`option indique le nom de fichier dans lequel le package téléchargé est écrit. Les options de l'exemple de commande suivant permettent de copier le fichier téléchargé dans le répertoire courant avec le nom local `awscliv2.zip`

     Pour spécifier une version, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *`2.0.30`* entraînerait la commande suivante : `awscli-exe-linux-x86_64-2.0.30.zip`

     ```
     $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o
      "awscliv2.zip"
     ```

     Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

   - Téléchargement depuis l'URL —

     Dans votre navigateur, téléchargez votre version spécifique du en AWS CLI ajoutant un trait d'union et le numéro de version au nom de fichier.

     ```
     https://awscli.amazonaws.com/awscli-exe-linux-x86_64-version.number.zip
     ```

     Dans cet exemple, le nom de fichier de la version *`2.0.30`* serait awscli-exe-linux-x 86_64-2.0.30.zip, d'où le lien suivant : https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip

Linux ARM

- Utiliser la **`curl`** commande — L'`-o`option indique le nom de fichier dans lequel le package téléchargé est écrit. Les options de l'exemple de commande suivant permettent de copier le fichier téléchargé dans le répertoire courant avec le nom local `awscliv2.zip`

  Pour spécifier une version, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *`2.0.30`* entraînerait la commande suivante : `awscli-exe-linux-aarch64-2.0.30.zip`

  ```
  $ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip" -o
   "awscliv2.zip"
  unzip awscliv2.zip
  sudo ./aws/install
  ```

- Téléchargement depuis l'URL —

  Dans votre navigateur, téléchargez votre version spécifique du en AWS CLI ajoutant un trait d'union et le numéro de version au nom de fichier.

  ```
  https://awscli.amazonaws.com/awscli-exe-linux-aarch64-version.number.zip
  ```

  Dans cet exemple, le nom de fichier de la version *`2.0.30`* donnerait `awscli-exe-linux-aarch64-2.0.30.zip` le lien suivant : https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip

2. (Facultatif) Vérification de l'intégrité du fichier zip que vous avez téléchargé

   Si vous avez choisi de télécharger manuellement le package AWS CLI d'installation `.zip` en suivant les étapes ci-dessus, vous pouvez utiliser les étapes suivantes pour vérifier les signatures à l'aide de l'GnuPGoutil.

   Les `.zip` fichiers du package AWS CLI d'installation sont signés cryptographiquement à l'aide de signatures PGP. En cas de dommage ou d'altération des fichiers, cette vérification échoue et vous ne devez pas poursuivre l'installation.

   a. Téléchargez et installez la commande gpg à l'aide de votre gestionnaire de packages. Pour de plus amples informations sur GnuPG, veuillez consulter le site Web GnuPG.

   b. Pour créer le fichier de clé publique, créez un fichier texte et collez le texte suivant.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgusOJl7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4ciIx5OuLlnJZIxSzx
PqGl0mkxImLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LblHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJOz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtWO
lrFj6UwAsGukBTAOxC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutlIqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBFqglXmZ7l9ywG
EEUJYOlb2XrSuPWml39beWdKM8kzr1OjnlOm6+lpTRCBfo0wa9F8YZRhHPAkwKkX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbGlAYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
aGveYQUJDMpiLAAKCRCmMQrMRnJHXKBYD/9Ab0qQdGiO5hObchG8xh8Rpb4Mjyf6
0JrVo6m8GNjNj6BHkSc8fuTQJ/FaEhaQxj3pjZ3GXPrXjIIVChmICLlFuRXYzrXc
Pw0lniybypsZEVai5kO0tCNBCCFuMN9RsmmRG8mf7lC4FSTbUDmxG/QlYK+0IV/l
uJkzxWa+rySkdpm0JdqumjegNRgObdXHAQDWlubWQHWyZyIQ2B4U7AxqSpcdJp6I
S4Zds4wVLd1WE5pquYQ8vS2cNlDm4QNg8wTj58e3lKN47hXHMIb6CHxRnb947oJa
pg189LLPR5koh+EorNkA1wu5mAJtJvy5YMsppy2y/kIjp3lyY6AmPT1posgGk70Z
CmToEZ5rbd7ARExtlh76A0cabMDFlEHDIK8RNUOSRr7L64+KxOUegKBfQHb9dADY
qqiKqpCbKgvtWlds909Ms74JBgr2KwZCSY1HaOxnIr4CY43QRqAq5YHOay/mU+6w
hhmdF18vpyK0vfkvvGresWtSXbag7Hkt3XjaEw76BzxQH21EBDqU8WJVjHgU6ru+
DJTs+SxgJbaT3hb/vyjlw0lK+hFfhWKRwgOXH8vqducF95NRSUxtS4fpqxWVaw3Q
V2OWSjbne99A5EPEySzryFTKbMGwaTlAwMCwYevt4YT6eb7NmFhTx0Fis4TalUs+
j+c7Kg92pDx2uQ==
=OBAt
-----END PGP PUBLIC KEY BLOCK-----
```

Pour référence, les détails de la clé publique sont les suivants.

```
Key ID:          A6310ACC4672
Type:            RSA
Size:            4096/4096
Created:         2019-09-18
Expires:         2026-07-07
User ID:         AWS CLI Team <aws-cli@amazon.com>
Key fingerprint:  FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

c. Importez la clé AWS CLI publique à l'aide de la commande suivante, en la *public-key-file-name* remplaçant par le nom de fichier de la clé publique que vous avez créée.

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
 imported
gpg: Total number processed: 1
gpg:               imported: 1
```

d. Téléchargez le fichier de AWS CLI signature du package que vous avez téléchargé. Celui-ci a le même chemin d'accès et le même nom que le fichier `.zip` auquel il correspond, mais son extension est `.sig`. Dans les exemples suivants, nous l'enregistrons dans le répertoire courant sous la forme d'un fichier nommé `awscliv2.sig`.

   Linux x86 (64-bit)

   Pour la dernière version de AWS CLI, utilisez le bloc de commande suivant :

   ```
   $ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-
   x86_64.zip.sig
   ```

   Pour une version spécifique de AWS CLI, ajoutez un trait d'union et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *2.0.30* entraînerait la commande suivante : `awscli-exe-linux-x86_64-2.0.30.zip.sig`

   ```
   $ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-
   x86_64-2.0.30.zip.sig
   ```

   Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

   Linux ARM

   Pour la dernière version de AWS CLI, utilisez le bloc de commande suivant :

   ```
   $ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-
   aarch64.zip.sig
   ```

   Pour une version spécifique de AWS CLI, ajoutez un trait d'union et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *2.0.30* entraînerait la commande suivante : `awscli-exe-linux-aarch64-2.0.30.zip.sig`

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-
aarch64-2.0.30.zip.sig
```

Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

e. Vérifiez la signature en transmettant les noms de fichiers `.sig` et `.zip` téléchargés sous forme de paramètres à la commande `gpg`.

```
$ gpg --verify awscliv2.sig awscliv2.zip
```

La sortie doit ressembler à ce qui suit :

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672
 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

> **Important**
>
> L'avertissement mentionné dans la sortie est attendu et n'indique pas de problème. Cela se produit parce qu'il n'existe pas de chaîne de confiance entre votre clé PGP personnelle (si vous en avez une) et la clé AWS CLI PGP. Pour de plus amples informations, consultez Web of trust.

3. Décompressez le programme d'installation. Si votre distribution Linux n'a pas de commande `unzip` intégrée, utilisez un équivalent pour procéder à la décompression. L'exemple de commande suivant décompresse le package et crée un répertoire nommé `aws` dans le répertoire courant.

```
$ unzip awscliv2.zip
```

4. Exécutez le programme d'installation. La commande d'installation utilise un fichier nommé `install` dans le répertoire `aws` récemment décompressé. Par défaut, les fichiers sont tous

installés dans `/usr/local/aws-cli` et un lien symbolique est créé dans `/usr/local/bin`. La commande inclut `sudo` pour accorder des autorisations d'écriture sur ces répertoires.

```
$ sudo ./aws/install
```

Vous pouvez effectuer l'installation sans `sudo` si vous spécifiez des répertoires sur lesquels vous disposez déjà d'autorisations en écriture. Utilisez les instructions suivantes pour la commande `install` de manière à spécifier l'emplacement d'installation :

- Assurez-vous que les chemins d'accès que vous fournissez aux paramètres `-i` et `-b` ne contiennent aucun nom de volume ou de répertoire incluant des espaces quelconques. Dans le cas contraire, l'installation échouera.
- `--install-dir`ou `-i` — Cette option indique le répertoire dans lequel copier tous les fichiers.

  La valeur par défaut est `/usr/local/aws-cli`.
- `--bin-dir`ou `-b` — Cette option indique que le `aws` programme principal du répertoire d'installation est symboliquement lié au fichier `aws` dans le chemin spécifié. Vous devez disposer des autorisations en écriture sur le répertoire spécifié. La création d'un lien symbolique vers un répertoire se trouvant déjà dans votre chemin d'accès vous dispense d'ajouter le répertoire d'installation à la variable `$PATH` de l'utilisateur.

  La valeur par défaut est `/usr/local/bin`.

```
$ ./aws/install -i /usr/local/aws-cli -b /usr/local/bin
```

> **Note**
>
> Pour mettre à jour votre installation actuelle de la AWS CLI version 2 vers une version plus récente, ajoutez votre lien symbolique et les informations d'installation existants pour créer la `install` commande avec le `--update` paramètre.
>
> ```
> $ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
> ```
>
> Pour localiser le lien symbolique et le répertoire d'installation existants, procédez comme suit :

1. Utilisez la commande `which` pour rechercher votre lien symbolique. Cela vous indique le chemin à utiliser avec le paramètre `--bin-dir`.

   ```
   $ which aws
   /usr/local/bin/aws
   ```

2. Utilisez la commande `ls` pour rechercher le répertoire vers lequel votre lien symbolique pointe. Cela vous indique le chemin à utiliser avec le paramètre `--install-dir`.

   ```
   $ ls -l /usr/local/bin/aws
   lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
   ```

5. Confirmez l'installation à l'aide de la commande suivante.

   ```
   $ aws --version
   aws-cli/2.27.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64
   ```

   Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

macOS

Exigences d'installation

- Vous savez quelle version de la AWS CLI version 2 vous souhaitez installer. Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub
- Nous prenons en charge la AWS CLI version 2 sur les versions de macOS 64 bits compatibles avec Apple.
- Comme elle AWS ne gère pas de référentiels tiers, nous ne pouvons pas garantir qu'ils contiennent la dernière version du AWS CLI.

Instructions d'installation

Vous pouvez installer la AWS CLI version 2 sur macOS de différentes manières.

GUI installer

Les étapes suivantes indiquent comment installer ou mettre à jour la dernière version de la AWS CLI version 2 à l'aide de l'interface utilisateur standard de macOS et de votre navigateur. Si vous effectuez une mise à jour vers la dernière version, utilisez la même méthode d'installation que celle utilisée pour la version actuelle.

1. Dans votre navigateur, téléchargez votre version spécifique du en AWS CLI ajoutant un trait d'union et le numéro de version au nom de fichier.

   ```
   https://awscli.amazonaws.com/AWSCLIV2-version.number.pkg
   ```

   Pour cet exemple, le nom de fichier de la version *2.0.30* `AWSCLIV2-2.0.30.pkg` entraînerait le lien suivant :https://awscli.amazonaws.com/AWSCLIV2-2.0.30.pkg.

2. Lancez le fichier téléchargé et suivez les instructions affichées à l'écran. Vous pouvez choisir d'installer la AWS CLI version 2 de la manière suivante :

   - Pour tous les utilisateurs de l'ordinateur (nécessite **sudo**)
     - Vous pouvez installer dans n'importe quel dossier ou choisir le dossier par défaut recommandé, à savoir `/usr/local/aws-cli`.
     - Le programme d'installation crée automatiquement un lien symbolique à l'adresse `/usr/local/bin/aws` qui fait le lien avec le programme principal du dossier d'installation que vous avez choisi.
   - Pour le seul utilisateur actuel (ne nécessite pas **sudo**)
     - Vous pouvez installer dans n'importe quel dossier sur lequel vous disposez d'une autorisation d'écriture.
     - En raison des autorisations utilisateur standard, une fois le programme d'installation terminé, vous devez créer manuellement un fichier de lien symbolique dans votre variable `$PATH` qui pointe vers les programmes `aws` et `aws_completer` en entrant les commandes suivantes à l'invite de commandes. Si votre variable `$PATH` inclut un dossier dans lequel vous pouvez écrire, vous pouvez exécuter la commande suivante sans `sudo` si vous spécifiez ce dossier comme chemin d'accès de la cible. Si vous n'avez pas de dossier accessible en écriture dans votre variable `$PATH`, vous devez utiliser `sudo` dans les commandes pour obtenir les autorisations d'écriture dans le dossier cible spécifié. L'emplacement par défaut d'un lien symbolique est `/usr/local/bin/`.

```
$ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
$ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
aws_completer
```

> **Note**
>
> Vous pouvez afficher les journaux de débogage pour l'installation en appuyant sur Cmd+L n'importe où dans le programme d'installation. Un volet du journal s'ouvre alors qui vous permet de filtrer et d'enregistrer le journal. Le fichier journal est également automatiquement enregistré dans `/var/log/install.log`.

3. Pour vérifier que le shell peut rechercher et exécuter la commande `aws` dans votre variable `$PATH`, utilisez les commandes suivantes.

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Darwin/23.3.0
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Command line installer - All users

Si vous êtes `sudo` autorisé, vous pouvez installer la AWS CLI version 2 pour tous les utilisateurs de l'ordinateur. Nous fournissons les étapes dans un groupe facile à copier et coller. Consultez les descriptions de chaque ligne dans les étapes suivantes.

Pour une version spécifique de AWS CLI, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *`2.0.30`* entraînerait la commande suivante : `AWSCLIV2-2.0.30.pkg`

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2-2.0.30.pkg" -o "AWSCLIV2.pkg"
$ sudo installer -pkg AWSCLIV2.pkg -target /
```

1. Vous pouvez télécharger le fichier à l'aide de la commande `curl`. L'option `-o` spécifie le nom de fichier dans lequel le package téléchargé est écrit. Dans cet exemple, le fichier est écrit dans `AWSCLIV2.pkg` du dossier actuel.

   Pour une version spécifique de AWS CLI, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *`2.0.30`* entraînerait la commande suivante : `AWSCLIV2-2.0.30.pkg`

   ```
   $ curl "https://awscli.amazonaws.com/AWSCLIV2-2.0.30.pkg" -o "AWSCLIV2.pkg"
   ```

   Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

2. Exécutez le programme `installer` macOS standard, en spécifiant le fichier `.pkg` téléchargé comme source. Utilisez le paramètre `-pkg` pour spécifier le nom du package à installer et le paramètre `-target /` pour spécifier l'unité sur laquelle le package doit être installé. Les fichiers sont installés dans `/usr/local/aws-cli`, et un lien symbolique est automatiquement créé dans `/usr/local/bin`. Vous devez inclure `sudo` dans la commande pour accorder les autorisations d'écriture à ces dossiers.

   ```
   $ sudo installer -pkg ./AWSCLIV2.pkg -target /
   ```

   Une fois l'installation terminée, les journaux de débogage sont écrits dans `/var/log/install.log`.

3. Pour vérifier que le shell peut rechercher et exécuter la commande `aws` dans votre variable `$PATH`, utilisez les commandes suivantes.

   ```
   $ which aws
   /usr/local/bin/aws
   $ aws --version
   aws-cli/2.27.41 Python/3.11.6 Darwin/23.3.0
   ```

   Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Command line - Current user

1. Pour spécifier le dossier dans lequel AWS CLI il est installé, vous devez créer un fichier XML. Ce fichier est un fichier au format XML qui ressemble à l'exemple suivant. Laissez toutes les valeurs comme indiqué, sauf que vous devez remplacer le chemin */Users/myusername* de la ligne 9 par le chemin du dossier dans lequel vous souhaitez installer la AWS CLI version 2. Le dossier doit déjà exister, sinon la commande échoue. Cet exemple XML indique que le programme d'installation installe le AWS CLI dans le dossier`/Users/myusername`, où il crée un dossier nomméaws-cli.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
  <array>
    <dict>
      <key>choiceAttribute</key>
      <string>customLocation</string>
      <key>attributeSetting</key>
      <string>/Users/myusername</string>
      <key>choiceIdentifier</key>
      <string>default</string>
    </dict>
  </array>
</plist>
```

2. Téléchargez le programme d'installation pkg à l'aide de la commande `curl`. L'option `-o` spécifie le nom de fichier dans lequel le package téléchargé est écrit. Dans cet exemple, le fichier est écrit dans `AWSCLIV2.pkg` du dossier actuel.

   Pour la version spécifique de AWS CLI, ajoutez un tiret et le numéro de version au nom du fichier. Dans cet exemple, le nom de fichier de la version *2.0.30* entraînerait la commande suivante : `AWSCLIV2-2.0.30.pkg`

   ```
   $ curl "https://awscli.amazonaws.com/AWSCLIV2-2.0.30.pkg" -o "AWSCLIV2.pkg"
   ```

   Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

3. Exécutez le programme `installer` macOS standard avec les options suivantes :

- Spécifiez le nom du package à installer à l'aide du paramètre `-pkg`.
- Spécifiez l'installation pour un utilisateur actuel uniquement en définissant le `-target` paramètre sur`CurrentUserHomeDirectory`.
- Spécifiez le chemin d'accès (relatif au dossier actuel) et le nom du fichier XML que vous avez créé dans le paramètre `-applyChoiceChangesXML`.

L'exemple suivant installe le AWS CLI dans le dossier. `/Users/myusername/aws-cli`

```
$ installer -pkg AWSCLIV2.pkg \
            -target CurrentUserHomeDirectory \
            -applyChoiceChangesXML choices.xml
```

4. Les autorisations utilisateur standard n'autorisant généralement pas l'écriture dans les dossiers de votre variable `$PATH`, le programme d'installation de ce mode n'essaie pas d'ajouter les liens symboliques aux programmes `aws` et `aws_completer`. AWS CLI Pour que le programme s'exécute correctement, vous devez créer manuellement les liens symboliques une fois le programme d'installation terminé. Si votre variable `$PATH` inclut un dossier dans lequel vous pouvez écrire et que vous spécifiez ce dossier comme chemin d'accès de la cible, vous pouvez exécuter la commande suivante sans `sudo`. Si vous ne disposez pas d'un dossier accessible en écriture dans votre variable `$PATH`, vous devez utiliser `sudo` pour les autorisations d'écriture dans le dossier cible spécifié. L'emplacement par défaut d'un lien symbolique est `/usr/local/bin/`.

```
$ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
$ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
aws_completer
```

Une fois l'installation terminée, les journaux de débogage sont écrits dans `/var/log/install.log`.

5. Pour vérifier que le shell peut rechercher et exécuter la commande `aws` dans votre variable `$PATH`, utilisez les commandes suivantes.

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Darwin/23.3.0
```

Si la `aws` commande est introuvable, vous devrez peut-être redémarrer votre terminal ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Windows

Exigences d'installation

- Vous savez quelle version de la AWS CLI version 2 vous souhaitez installer. Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub
- Nous prenons en charge les versions AWS CLI de Windows 64 bits prises en charge par Microsoft.
- Droits d'administrateur pour installer le logiciel

Instructions d'installation

Pour mettre à jour votre installation actuelle de la AWS CLI version 2 sous Windows, téléchargez un nouveau programme d'installation à chaque mise à jour pour remplacer les versions précédentes. AWS CLI est régulièrement mis à jour. Pour savoir quand la dernière version a été publiée, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

1. Téléchargez et exécutez le programme d'installation AWS CLI MSI pour Windows (64 bits) de l'une des manières suivantes :

   - Téléchargement et exécution du programme d'installation MSI : pour créer votre lien de téléchargement pour une version spécifique du AWS CLI, ajoutez un tiret et le numéro de version au nom du fichier.

     ```
     https://awscli.amazonaws.com/AWSCLIV2-version.number.msi
     ```

     Pour cet exemple, le nom de fichier de la version *2.0.30* entraînerait le lien suivant :https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi. `AWSCLIV2-2.0.30.msi`
   - À l'aide de la commande msiexec : vous pouvez également utiliser le programme d'installation MSI en ajoutant le lien vers la commande. `msiexec` Pour une version spécifique de AWS CLI, ajoutez un tiret et le numéro de version au nom du fichier.

     ```
     C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2-version.number.msi
     ```

Dans cet exemple, le nom de fichier de la version *2.0.30* donnerait `AWSCLIV2-2.0.30.msi` le lien suivanthttps://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi
```

Pour les différents paramètres pouvant être utilisés avec`msiexec`, consultez msiexec sur le site Web de Microsoft Docs.

Pour une liste des versions, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

2. Pour confirmer l'installation, ouvrez le menu Démarrer, recherchez `cmd` pour ouvrir une fenêtre d'invite de commandes et, à l'invite de commandes, utilisez la commande `aws --version`.

```
C:\> aws --version
aws-cli/2.27.41 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

Si Windows ne trouve pas le programme, vous devrez peut-être fermer puis rouvrir la fenêtre d'invite de commande pour actualiser le chemin, ou suivre les instructions de résolution des problèmes dansthe section called “Résoudre les erreurs”.

Résolution des erreurs AWS CLI d'installation et de désinstallation

Si vous rencontrez des problèmes après l'installation ou la désinstallation du AWS CLI, consultez les étapes de résolution the section called “Résoudre les erreurs” des problèmes. Pour connaître les étapes de résolution des problèmes les plus pertinentesthe section called “Erreurs de commande non trouvées”, reportez-vous aux the section called “La commande « `aws --version` » renvoie une version différente de celle que vous avez installée” sections, etthe section called “La commande « `aws --version` » renvoie une version après la désinstallation du AWS CLI”.

Étapes suivantes

Après avoir terminé les étapes the section called “Prérequis” et installé le AWS CLI, vous devez effectuer unthe section called “Configuration”.

Construire et installer le AWS CLI depuis le code source

Cette rubrique décrit comment installer ou mettre à jour depuis le code source vers la dernière version de AWS Command Line Interface (AWS CLI) sur les systèmes d'exploitation pris en charge.

Pour plus d'informations sur les dernières versions de AWS CLI, consultez le journal des modifications de la AWS CLI version 2 sur. GitHub

> **Important**
>
> AWS CLI les versions 1 et 2 utilisent le même nom de `aws` commande. Si vous avez déjà installé AWS CLI la version 1, consultezGuide de migration pour la AWS CLI version 2.

Rubriques

- Pourquoi créer à partir des sources ?
- Étapes rapides
- Étape 1 : Configuration de toutes les exigences
- Étape 2 : Configuration de l'installation AWS CLI source
- Étape 3 : Construire le AWS CLI
- Étape 4 : Installation du AWS CLI
- Étape 5 : vérification de l' AWS CLI installation
- Exemples de flux de travail
- Résolution des erreurs AWS CLI d'installation et de désinstallation
- Étapes suivantes

Pourquoi créer à partir des sources ?

AWS CLI Il est disponible sous forme d'installateurs prédéfinis pour la plupart des plateformes et environnements, ainsi que sous forme d'image Docker.

En général, ces installateurs fournissent une couverture pour la plupart des cas d'utilisation. Les instructions d'installation à partir des sources ont pour but de vous aider à résoudre les cas d'utilisation que nos installateurs ne couvrent pas. Certains de ces cas d'utilisation sont les suivants :

- Les programmes d'installation prédéfinis ne sont pas compatibles avec votre environnement. Par exemple, ARM 32 bits n'est pas pris en charge par les programmes d'installation prédéfinis.
- Les programmes d'installation prédéfinis comportent des dépendances qui font défaut à votre environnement. Par exemple, Alpine Linux l'utilise `musl`, mais les programmes d'installation actuels nécessitent que `glibc` les programmes d'installation prédéfinis ne fonctionnent pas immédiatement.
- Les installateurs prédéfinis nécessitent des ressources auxquelles votre environnement restreint l'accès. Par exemple, les systèmes de sécurité renforcés peuvent ne pas autoriser l'accès à la mémoire partagée. Cela est nécessaire pour le programme d'`aws`installation gelé.
- Les installateurs prédéfinis sont souvent des bloqueurs pour les responsables des gestionnaires de paquets, car il est préférable de contrôler totalement le processus de création du code et des packages. La création à partir des sources permet aux responsables de la distribution de rationaliser le processus pour maintenir les AWS CLI mises à jour. L'activation des responsables fournit aux clients davantage de up-to-date versions du AWS CLI lors de l'installation à partir d'un gestionnaire de packages tiers tel que `brew`yum, et. `apt`
- Les clients qui utilisent des AWS CLI fonctionnalités de correctif doivent les créer et les installer AWS CLI depuis le code source. Cela est particulièrement important pour les membres de la communauté qui souhaitent tester les modifications qu'ils ont apportées à la source avant de les apporter au AWS CLI GitHub dépôt.

Étapes rapides

> **Note**
>
> Tous les exemples de code sont supposés s'exécuter à partir de la racine du répertoire source.

Pour créer et installer le AWS CLI depuis le code source, suivez les étapes décrites dans cette section. Il AWS CLI utilise GNU Autotools pour l'installer à partir des sources. Dans le cas le plus simple, AWS CLI vous pouvez l'installer à partir de la source en exécutant les exemples de commandes par défaut depuis la racine du AWS CLI GitHub référentiel.

1. Configurez toutes les exigences pour votre environnement. Cela inclut la possibilité d'exécuter des fichiers générés par GNU Autotools et d'installer Python 3.8 ou une version ultérieure.

2. Dans votre terminal, naviguez jusqu'au niveau supérieur du dossier AWS CLI source et exécutez la `./configure` commande. Cette commande vérifie le système pour toutes les dépendances requises et génère un fichier `Makefile` pour la création et l'installation AWS CLI en fonction des configurations détectées et spécifiées.

   Linux and macOS

   L'exemple de `./configure` commande suivant définit la configuration de construction pour l' AWS CLI utilisation des paramètres par défaut.

   ```
   $ ./configure
   ```

   Windows PowerShell

   Avant d'exécuter un appel de commande MSYS2, vous devez conserver votre répertoire de travail actuel :

   ```
   PS C:\> $env:CHERE_INVOKING = 'yes'
   ```

   Utilisez ensuite l'exemple de `./configure` commande suivant pour définir la configuration de compilation afin d' AWS CLI utiliser votre chemin local vers votre exécutable Python, de l'installer dans C:\Program Files \ AWSCLI et de télécharger toutes les dépendances.

   ```
   PS C:\> C:\msys64\usr\bin\bash  -lc " PYTHON='C:\path\to\python.exe' ./configure
    --prefix='C:\Program Files\AWSCLI' --with-download-deps "
   ```

   Pour plus de détails, les options de configuration disponibles et les informations sur les paramètres par défaut, consultez la the section called “Étape 2 : Configuration de l'installation AWS CLI source ” section.

3. Exécutez la commande `make`. Cette commande crée le AWS CLI en fonction de vos paramètres de configuration.

   L'exemple de `make` commande suivant utilise les options par défaut en utilisant vos `./configure` paramètres existants.

Linux and macOS

```
$ make
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "make"
```

Pour plus de détails et les options de compilation disponibles, consultez la the section called “Étape 3 : Construire le AWS CLI” section.

4. Exécutez la commande `make install`. Cette commande installe votre build AWS CLI à l'emplacement configuré sur votre système.

   L'exemple de `make install` commande suivant installe votre build AWS CLI et crée des liens symboliques dans les emplacements que vous avez configurés à l'aide des paramètres de commande par défaut.

   Linux and macOS

   ```
   $ make install
   ```

   Windows PowerShell

   ```
   PS C:\> C:\msys64\usr\bin\bash -lc "make install"
   ```

   Après l'installation, ajoutez le chemin à l' AWS CLI aide de ce qui suit :

   ```
   PS C:\> $Env:PATH +=";C:\Program Files\AWSCLI\bin\"
   ```

   Pour plus de détails et les options d'installation disponibles, consultez la the section called “Étape 4 : Installation du AWS CLI” section.

5. Confirmez l'installation AWS CLI avec succès à l'aide de la commande suivante :

   ```
   $ aws --version
   aws-cli/2.27.41 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
   ```

Pour les étapes de résolution des erreurs d'installation, consultez la the section called “Résolution des erreurs AWS CLI d'installation et de désinstallation” section.

Étape 1 : Configuration de toutes les exigences

Pour créer le fichier AWS CLI à partir des sources, vous devez au préalable remplir les conditions suivantes :

> **Note**
>
> Tous les exemples de code sont supposés s'exécuter à partir de la racine du répertoire source.

1. Téléchargez le AWS CLI code source en bifurquant le AWS CLI GitHub dépôt ou en téléchargeant l'archive source. Les instructions sont l'une des suivantes :

   - Forker et cloner le AWS CLI dépôt à partir de GitHub. Pour plus d'informations, voir Forker un dépôt dans la GitHub documentation.
   - Téléchargez la dernière archive source pour https://awscli.amazonaws.com/awscli.tar.gz extraire le contenu à l'aide des commandes suivantes :

   ```
   $ curl "https://awscli.amazonaws.com/awscli.tar.gz" -o "awscli.tar.gz"
   $ tar -xzf awscli.tar.gz
   ```

   > **Note**
   >
   > Pour télécharger une version spécifique, utilisez le format de lien suivant : https://awscli.amazonaws.com*versionnumber*/awscli-.tar.gz
   > Par exemple, pour la version 2.10.0, le lien est le suivant : https://awscli.amazonaws.com /awscli- .tar.gz *2.10.0*
   > Les versions source sont disponibles à partir de la version 2.10.0 du. AWS CLI

   (Facultatif) Vérifiez l'intégrité du fichier zip que vous avez téléchargé en effectuant les étapes suivantes :

1. Vous pouvez utiliser les étapes suivantes pour vérifier les signatures à l'aide de l'outil GnuPG.

   Les `.zip` fichiers du package AWS CLI d'installation sont signés cryptographiquement à l'aide de signatures PGP. En cas de dommage ou d'altération des fichiers, cette vérification échoue et vous ne devez pas poursuivre l'installation.

2. Téléchargez et installez la commande gpg à l'aide de votre gestionnaire de packages. Pour de plus amples informations sur GnuPG, veuillez consulter le site Web GnuPG.

3. Pour créer le fichier de clé publique, créez un fichier texte et collez le texte suivant.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgusOJl7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4ciIx5OuLlnJZIxSzx
PqGl0mkxImLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LblHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJOz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtWO
lrFj6UwAsGukBTAOxC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutlIqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBFqglXmZ7l9ywG
EEUJYOlb2XrSuPWml39beWdKM8kzr1OjnlOm6+lpTRCBfo0wa9F8YZRhHPAkwKkX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbGlAYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
aGveYQUJDMpiLAAKCRCmMQrMRnJHXKBYD/9Ab0qQdGiO5hObchG8xh8Rpb4Mjyf6
0JrVo6m8GNjNj6BHkSc8fuTQJ/FaEhaQxj3pjZ3GXPrXjIIVChmICLlFuRXYzrXc
Pw0lniybypsZEVai5kO0tCNBCCFuMN9RsmmRG8mf7lC4FSTbUDmxG/QlYK+0IV/l
uJkzxWa+rySkdpm0JdqumjegNRgObdXHAQDWlubWQHWyZyIQ2B4U7AxqSpcdJp6I
S4Zds4wVLd1WE5pquYQ8vS2cNlDm4QNg8wTj58e3lKN47hXHMIb6CHxRnb947oJa
pg189LLPR5koh+EorNkA1wu5mAJtJvy5YMsppy2y/kIjp3lyY6AmPT1posgGk70Z
CmToEZ5rbd7ARExtlh76A0cabMDFlEHDIK8RNUOSRr7L64+KxOUegKBfQHb9dADY
qqiKqpCbKgvtWlds909Ms74JBgr2KwZCSY1HaOxnIr4CY43QRqAq5YHOay/mU+6w
hhmdF18vpyK0vfkvvGresWtSXbag7Hkt3XjaEw76BzxQH21EBDqU8WJVjHgU6ru+
DJTs+SxgJbaT3hb/vyjlw0lK+hFfhWKRwgOXH8vqducF95NRSUxtS4fpqxWVaw3Q
V2OWSjbne99A5EPEySzryFTKbMGwaTlAwMCwYevt4YT6eb7NmFhTx0Fis4TalUs+
j+c7Kg92pDx2uQ==
=OBAt
-----END PGP PUBLIC KEY BLOCK-----
```

   Pour référence, les détails de la clé publique sont les suivants.

```
Key ID:           A6310ACC4672
Type:             RSA
Size:             4096/4096
Created:          2019-09-18
Expires:          2026-07-07
User ID:          AWS CLI Team <aws-cli@amazon.com>
Key fingerprint:  FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

4. Importez la clé AWS CLI publique à l'aide de la commande suivante, en la *public-key-file-name* remplaçant par le nom de fichier de la clé publique que vous avez créée.

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
 imported
gpg: Total number processed: 1
gpg:               imported: 1
```

5. Téléchargez le fichier de AWS CLI signature du package que vous avez téléchargé sur https://awscli.amazonaws.com/awscli.tar.gz.sig. Il a le même chemin et le même nom que le fichier tarball auquel il correspond, mais porte l'extension `.sig`. Enregistrez-le dans le même chemin que le fichier tarball. Ou utilisez le bloc de commande suivant :

```
$ curl awscliv2.sig https://awscli.amazonaws.com/ -o awscli.tar.gz.sig
```

6. Vérifiez la signature en transmettant les noms de fichiers `.sig` et `.zip` téléchargés sous forme de paramètres à la commande gpg.

```
$ gpg --verify awscliv2.sig awscli.tar.gz
```

La sortie doit ressembler à ce qui suit :

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672
 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

> **Important**
>
> L'avertissement mentionné dans la sortie est attendu et n'indique pas de problème. Cela se produit parce qu'il n'existe pas de chaîne de confiance entre votre clé PGP personnelle (si vous en avez une) et la clé AWS CLI PGP. Pour de plus amples informations, consultez Web of trust.

2. Vous disposez d'un environnement capable d'exécuter des fichiers générés par GNU Autotools tels que `configure` et. `Makefile` Ces fichiers sont largement portables sur les plateformes POSIX.

   Linux and macOS

   Si Autotools n'est pas encore installé dans votre environnement ou si vous devez le mettre à jour, suivez les instructions d'installation figurant dans Comment installer les Autotools (en tant qu'utilisateur) ? ou Installation de base dans la documentation GNU.

   Windows PowerShell

   > **Warning**
   >
   > Si vous travaillez dans un environnement Windows, nous vous conseillons d'utiliser les programmes d'installation prédéfinis. Pour les instructions d'installation sur les programmes d'installation prédéfinis, voir the section called “Installation/mise à jour”

   Windows n'étant pas fourni avec un shell compatible POSIX, vous devez installer un logiciel supplémentaire pour installer le shell AWS CLI depuis le code source. MSYS2fournit un ensemble d'outils et de bibliothèques pour aider à créer et à installer des logiciels Windows, en particulier pour les scripts basés sur POSIX utilisés par Autotools.

   1. Installez MSYS2. Pour plus d'informations sur l'installation et l'utilisation MSYS2, consultez les instructions d'installation et d'utilisation dans la MSYS2 documentation.
   2. Ouvrez le MSYS2 terminal et installez les outils automatiques à l'aide de la commande suivante.

      ```
      $ pacman -S autotools
      ```

> **Note**
>
> Lorsque vous utilisez les exemples de code de configuration, de génération et d'installation présentés dans ce guide pour Windows, le chemin d' MSYS2 installation par défaut `C:\msys64\usr\bin\bash` est supposé. Lorsque vous appelez MSYS2 inside of, PowerShell vous utiliserez le format suivant, avec la commande bash entre guillemets :
>
> ```
> PS C:\> C:\msys64\usr\bin\bash  -lc "command example"
> ```
>
> L'exemple de commande suivant appelle la `./configure` commande.
>
> ```
> PS C:\> C:\msys64\usr\bin\bash  -lc "./configure"
> ```

3. Un interpréteur Python 3.8 ou version ultérieure est installé. La version minimale de Python requise suit les mêmes délais que la politique officielle de support de Python pour AWS SDKs et les outils. Un interprète n'est pris en charge que 6 mois après sa end-of-support date.
4. (Facultatif) Installez toutes les dépendances de la bibliothèque Python de compilation et d'exécution du AWS CLI. La `./configure` commande vous indique s'il vous manque des dépendances et vous indique comment les installer.

   Vous pouvez installer et utiliser automatiquement ces dépendances par le biais de la configuration, voir the section called “Téléchargement des dépendances” pour plus d'informations.

Étape 2 : Configuration de l'installation AWS CLI source

La configuration pour la création et l'installation du AWS CLI est spécifiée à l'aide du `configure` script. Pour la documentation de toutes les options de configuration, exécutez le `configure` script avec l'`--help`option :

Linux and macOS

```
$ ./configure --help
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure --help"
```

Les options les plus importantes sont les suivantes :

- Emplacement de l'installation
- Interpréteur Python
- Téléchargement des dépendances
- Type d'installation

Emplacement de l'installation

L'installation source du AWS CLI utilise deux répertoires configurables pour installer AWS CLI :

- `libdir`- Répertoire parent dans lequel le AWS CLI sera installé. Le chemin d'accès à l' AWS CLI installation est*<libdir-value>*`/aws-cli`. La `libdir` valeur par défaut pour Linux et macOS est de définir `/usr/local/lib` le répertoire d'installation par défaut. */usr/local/lib*`/aws-cli`
- `bindir`- Répertoire dans lequel les AWS CLI exécutables sont installés. L'emplacement par défaut est `/usr/local/bin`.

Les `configure` options suivantes contrôlent les répertoires utilisés :

- `--prefix`- Définit le préfixe de répertoire à utiliser pour l'installation. La valeur par défaut pour Linux et macOS est`/usr/local`.
- `--libdir`- Définit le `libdir` à utiliser pour installer le AWS CLI. La valeur par défaut est `<prefix-value>/lib`. Si `--libdir` les deux ne `--prefix` sont pas spécifiés, la valeur par défaut pour Linux et macOS est`/usr/local/lib/`.
- `--bindir`- Définit le `bindir` à utiliser pour installer les `aws_completer` exécutables AWS CLI `aws` et. La valeur par défaut est `<prefix-value>/bin`. Si `bindir` les deux ne `--prefix` sont pas spécifiés, la valeur par défaut pour Linux et macOS est`/usr/local/bin/`.

Linux and macOS

L'exemple de commande suivant utilise l'`--prefix`option permettant d'installer le AWS CLI. Cette commande installe le fichier AWS CLI in `$HOME/.local/lib/aws-cli` et les exécutables dans : `$HOME/.local/bin`

```
$ ./configure --prefix=$HOME/.local
```

L'exemple de commande suivant utilise l'`--libdir`option permettant d'installer l'application en AWS CLI tant qu'application complémentaire dans le `/opt` répertoire. Cette commande installe le fichier AWS CLI at `/opt/aws-cli` et les exécutables à leur emplacement par défaut de. `/usr/local/bin`

```
$ ./configure --libdir=/opt
```

Windows PowerShell

L'exemple de commande suivant utilise l'`--prefix`option permettant d'installer le AWS CLI. Cette commande installe le fichier AWS CLI in `$HOME/.local/lib/aws-cli` et les exécutables dans : `$HOME/.local/bin`

```
$ C:\msys64\usr\bin\bash -lc "./configure --prefix='C:\Program Files\AWSCLI'"
```

L'exemple de commande suivant utilise l'`--libdir`option permettant d'installer l'application en AWS CLI tant qu'application complémentaire dans le `/opt` répertoire. Cette commande installe le fichier AWS CLI at. `C:\Program Files\AWSCLI\opt\aws-cli`

Interpréteur Python

> **Note**
>
> Il est vivement recommandé de spécifier l'interpréteur Python lors de l'installation pour Windows.

Le `./configure` script sélectionne automatiquement un interpréteur Python 3.8 ou version ultérieure installé à utiliser pour créer et exécuter la macro à l' AWS CLI aide de la macro `AM_PATH_PYTHON`Autoconf.

L'interpréteur Python à utiliser peut être défini explicitement à l'aide de la variable d'PYTH0Nenvironnement lors de l'exécution du `configure` script :

Linux and macOS

```
$ PYTHON=/path/to/python ./configure
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "PYTHON='C:\path\to\python' ./configure"
```

Téléchargement des dépendances

Par défaut, il est nécessaire que toutes les dépendances de build et d'exécution du AWS CLI soient déjà installées sur le système. Cela inclut toutes les dépendances des bibliothèques Python. Toutes les dépendances sont vérifiées lors de l'exécution du `configure` script, et si aucune dépendance Python n'est absente du système, le `configure` script émet une erreur.

L'exemple de code suivant génère des erreurs lorsque des dépendances sont absentes de votre système :

Linux and macOS

```
$ ./configure
checking for a Python interpreter with version >= 3.8... python
checking for python... /Users/username/.envs/env3.11/bin/python
checking for python version... 3.11
checking for python platform... darwin
checking for GNU default python prefix... ${prefix}
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.11/
site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/
lib/python3.11/site-packages
checking for --with-install-type... system-sandbox
checking for --with-download-deps... Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 125, in <module>
    main()
```

```
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 121, in main
    parsed_args.func(parsed_args)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 49, in validate
    validate_env(parsed_args.artifact)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/validate_env.py",
 line 68, in validate_env
    raise UnmetDependenciesException(unmet_deps, in_venv)
validate_env.UnmetDependenciesException: Environment requires following Python
 dependencies:

awscrt (required: ('>=0.12.4', '<0.17.0')) (version installed: None)

We recommend using --with-download-deps flag to automatically create a virtualenv
 and download the dependencies.

If you want to manage the dependencies yourself instead, run the following pip
 command:
/Users/username/.envs/env3.11/bin/python -m pip install --prefer-binary
 'awscrt>=0.12.4,<0.17.0'

configure: error: "Python dependencies not met."
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure"
checking for a Python interpreter with version >= 3.8... python
checking for python... /Users/username/.envs/env3.11/bin/python
checking for python version... 3.11
checking for python platform... darwin
checking for GNU default python prefix... ${prefix}
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.11/
site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/
lib/python3.11/site-packages
checking for --with-install-type... system-sandbox
checking for --with-download-deps... Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 125, in <module>
```

```
    main()
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 121, in main
    parsed_args.func(parsed_args)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 49, in validate
    validate_env(parsed_args.artifact)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/validate_env.py",
 line 68, in validate_env
    raise UnmetDependenciesException(unmet_deps, in_venv)
validate_env.UnmetDependenciesException: Environment requires following Python
 dependencies:

awscrt (required: ('>=0.12.4', '<0.17.0')) (version installed: None)

We recommend using --with-download-deps flag to automatically create a virtualenv
 and download the dependencies.

If you want to manage the dependencies yourself instead, run the following pip
 command:
/Users/username/.envs/env3.11/bin/python -m pip install --prefer-binary
 'awscrt>=0.12.4,<0.17.0'

configure: error: "Python dependencies not met."
```

Pour installer automatiquement les dépendances Python requises, utilisez l'`--with-download-deps`option. Lorsque vous utilisez cet indicateur, le processus de génération effectue les opérations suivantes :

- Ignore la vérification des dépendances de la bibliothèque Python.
- Configure les paramètres pour télécharger toutes les dépendances Python requises et utiliser uniquement les dépendances téléchargées pour les créer AWS CLI pendant la `make` construction.

L'exemple de commande de configuration suivant utilise l'`--with-download-deps`option permettant de télécharger et d'utiliser les dépendances Python :

Linux and macOS

```
$ ./configure --with-download-deps
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure --with-download-deps"
```

Type d'installation

Le processus d'installation source prend en charge les types d'installation suivants :

- `system-sandbox`- (Par défaut) Crée un environnement virtuel Python isolé, l'installe AWS CLI dans l'environnement virtuel et crée des liens symboliques vers le `aws` `aws_completer` fichier exécutable dans l'environnement virtuel. Cette installation de AWS CLI dépend directement de l'interpréteur Python sélectionné pour son exécution.

  Il s'agit d'un mécanisme d'installation léger qui permet de l' AWS CLI installer sur un système et qui suit les meilleures pratiques de Python en sandboxant l'installation dans un environnement virtuel. Cette installation est destinée aux clients qui souhaitent installer le fichier à AWS CLI partir du code source de la manière la plus fluide possible avec l'installation couplée à votre installation de Python.
- `portable-exe`- Gèle le fichier AWS CLI dans un exécutable autonome qui peut être distribué dans des environnements d'architectures similaires. Il s'agit du même processus que celui utilisé pour générer les exécutables prédéfinis officiels du. AWS CLI Il `portable-exe` se bloque dans une copie de l'interpréteur Python choisi à l'`configure`étape à utiliser pour l'exécution du AWS CLI. Cela permet de le déplacer vers d'autres machines qui ne disposent peut-être pas d'un interpréteur Python.

  Ce type de build est utile car vous pouvez vous assurer que votre AWS CLI installation n'est pas couplée à la version Python installée dans l'environnement et vous pouvez distribuer une version sur un autre système sur lequel Python n'est peut-être pas déjà installé. Cela vous permet de contrôler les dépendances et la sécurité des AWS CLI exécutables que vous utilisez.

Pour configurer le type d'installation, utilisez l'`--with-install-type`option et spécifiez la valeur `portable-exe` ou`system-sandbox`.

L'exemple de `./configure` commande suivant spécifie une valeur de `portable-exe` :

Linux and macOS

```
$ ./configure --with-install-type=portable-exe
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure --with-install-type=portable-exe"
```

Étape 3 : Construire le AWS CLI

Utilisez la `make` commande pour créer le à l' AWS CLI aide de vos paramètres de configuration :

Linux and macOS

```
$ make
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "make"
```

> **Note**
>
> Lorsque vous utilisez la **`make`** commande, les étapes suivantes sont effectuées en arrière-plan :
>
> 1. Un environnement virtuel est créé dans le répertoire de construction à l'aide du `venv`module Python. L'environnement virtuel est amorcé avec une version de pip fournie dans la bibliothèque standard Python.
> 2. Copie les dépendances de la bibliothèque Python. Selon que l'`--with-download-deps`indicateur est spécifié dans la `configure` commande, cette étape effectue l'une des opérations suivantes :
>
>   - Le `--with-download-deps` est spécifié. Les dépendances Python sont installées par pip. Cela inclut `wheel`, `setuptools`, et toutes les dépendances AWS CLI d'exécution. Si vous construisez le`portable-exe`, `pyinstaller` est installé. Ces exigences sont toutes spécifiées dans les fichiers de verrouillage générés à partir de `pip-compile`.

- Le n'**`--with-download-deps`**est pas spécifié. Les bibliothèques Python du package du site de l'interpréteur Python ainsi que tous les scripts (par exemple`pyinstaller`) sont copiés dans l'environnement virtuel utilisé pour la construction.

3. S'exécute `pip install` directement sur la AWS CLI base de code pour effectuer une compilation hors ligne dans l'arborescence et l'installer AWS CLI dans l'environnement virtuel de génération. Cette installation utilise les drapeaux pip -- no-build-isolation , -- use-feature= in-tree-build , -- et. no-cache-dir `--no-index`
4. (Facultatif) Si le `--install-type` paramètre est défini sur `portable-exe` dans la `configure` commande, crée un exécutable autonome à l'aide `pyinstaller`de.

Étape 4 : Installation du AWS CLI

La `make install` commande installe votre build AWS CLI à l'emplacement configuré sur le système.

Linux and macOS

L'exemple de commande suivant installe le à l' AWS CLI aide de vos paramètres de configuration et de compilation :

```
$ make install
```

Windows PowerShell

L'exemple de commande suivant installe le à l' AWS CLI aide de vos paramètres de configuration et de construction, puis ajoute une variable d'environnement avec le chemin pour : AWS CLI

```
PS C:\> C:\msys64\usr\bin\bash -lc " make install "
PS C:\> $Env:PATH +=";C:\Program Files\AWSCLI\bin\"
```

La `make install` règle prend en charge la `DESTDIR`variable. Lorsqu'elle est spécifiée, cette variable préfixe le chemin spécifié au chemin d'installation déjà configuré lors de l'installation du AWS CLI. Par défaut, aucune valeur n'est définie pour cette variable.

Linux and macOS

L'exemple de code suivant utilise un `--prefix=/usr/local` indicateur pour configurer un emplacement d'installation, puis modifie cette destination à l'aide `DESTDIR=/tmp/stage` de la `make install` commande. Ces commandes entraînent l'installation AWS CLI sur `/tmp/stage/usr/local/lib/aws-cli` et la localisation de ses exécutables dans. `/tmp/stage/usr/local/bin`

```
$ ./configure --prefix=/usr/local
$ make
$ make DESTDIR=/tmp/stage install
```

Windows PowerShell

L'exemple de code suivant utilise un `--prefix=\awscli` indicateur pour configurer un emplacement d'installation, puis modifie cette destination à l'aide `DESTDIR=C:\Program Files` de la `make install` commande. Ces commandes entraînent l' AWS CLI installation sur`C:\Program Files\awscli`.

```
$ ./configure --prefix=\awscli
$ make
$ make DESTDIR='C:\Program Files' install
```

> **Note**
>
> Lors de l'exécution**`make install`**, les étapes suivantes sont effectuées en arrière-plan
>
> 1. Déplace l'un des éléments suivants vers le répertoire d'installation configuré :
>   - Si le type d'installation est`system-sandbox`, déplace votre environnement virtuel créé.
>   - Si le type d'installation est un`portable-exe`, déplace votre exécutable autonome créé.
> 2. Crée des liens symboliques pour les `aws_completer` exécutables `aws` et dans le répertoire bin que vous avez configuré.

Étape 5 : vérification de l' AWS CLI installation

Confirmez l'installation AWS CLI avec succès à l'aide de la commande suivante :

```
$ aws --version
aws-cli/2.27.41 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

Si la `aws` commande n'est pas reconnue, vous devrez peut-être redémarrer votre terminal pour mettre à jour les nouveaux liens symboliques. Si vous rencontrez des problèmes supplémentaires après l'installation ou la désinstallation du AWS CLI, consultez les étapes de dépannage the section called “Résoudre les erreurs” les plus courantes

Exemples de flux de travail

Cette section fournit des exemples de flux de travail de base pour l'installation à partir des sources.

Installation de base de Linux et de macOS

L'exemple suivant est un flux de travail d'installation de base dans lequel le AWS CLI est installé à l'emplacement par défaut de`/usr/local/lib/aws-cli`.

```
$ cd path/to/cli/respository/
$ ./configure
$ make
$ make install
```

Installation automatique de Windows

> **Note**
>
> Vous devez vous présenter PowerShell en tant qu'administrateur pour utiliser ce flux de travail.

MSYS2 peut être utilisé de manière automatisée dans un paramètre CI, voir Utilisation MSYS2 dans CI dans la MSYS2 documentation.

Downloaded Tarball

Téléchargez le `awscli.tar.gz` fichier, extrayez-le et installez le AWS CLI. Lorsque vous utilisez les commandes suivantes, remplacez les chemins suivants :

- `C:\msys64\usr\bin\bash`avec l'emplacement de votre MSYS2 chemin.
- `.\awscli-2.x.x\`avec le nom du `awscli.tar.gz` dossier extrait.
- `PYTHON='C:\path\to\python.exe'`avec votre chemin Python local.

L'exemple de code suivant automatise la création et l'installation AWS CLI du formulaire à PowerShell l'aide MSYS2, et indique l'installation locale de Python à utiliser :

```
PS C:\> curl "https://awscli.amazonaws.com/awscli.tar.gz" -o "awscliv2.zip"  #
 Download the awscli.tar.gz file in the current working directory
PS C:\> tar -xvzf .\awscli.tar.gz #  Extract awscli.tar.gz file
PS C:\> cd .\awscli-2.x.x\ #  Navigate to the root of the extracted files
PS C:\> $env:CHERE_INVOKING = 'yes' #  Preserve the current working directory
PS C:\> C:\msys64\usr\bin\bash  -lc " PYTHON='C:\path\to\python.exe' ./configure --
prefix='C:\Program Files\AWSCLI' --with-download-deps "
PS C:\> C:\msys64\usr\bin\bash  -lc "make"
PS C:\> C:\msys64\usr\bin\bash  -lc "make install"
PS C:\> $Env:PATH +=";C:\Program Files\AWSCLI\bin\"
PS C:\> aws --version
aws-cli/2.27.41 Python/3.11.6 Windows/10 source-sandbox/AMD64
```

GitHub Repository

Téléchargez le `awscli.tar.gz` fichier, extrayez-le et installez le AWS CLI. Lorsque vous utilisez les commandes suivantes, remplacez les chemins suivants :

- `C:\msys64\usr\bin\bash`avec l'emplacement de votre MSYS2 chemin.
- `C:path\to\cli\repository\`avec le chemin d'accès à votre AWS CLI dépôt cloné depuis GitHub. Pour plus d'informations, voir Forker un dépôt dans la documentation GitHub
- `PYTHON='C:\path\to\python.exe'`avec votre chemin Python local.

L'exemple de code suivant automatise la création et l'installation AWS CLI du formulaire à PowerShell l'aide MSYS2, et indique l'installation locale de Python à utiliser :

```
PS C:\> cd C:path\to\cli\repository\
```

```
PS C:\> $env:CHERE_INVOKING = 'yes' #  Preserve the current working directory
PS C:\> C:\msys64\usr\bin\bash  -lc " PYTHON='C:\path\to\python.exe' ./configure --
prefix='C:\Program Files\AWSCLI' --with-download-deps "
PS C:\> C:\msys64\usr\bin\bash  -lc "make"
PS C:\> C:\msys64\usr\bin\bash  -lc "make install"
PS C:\> $Env:PATH +=";C:\Program Files\AWSCLI\bin\"
PS C:\> aws --version
```

Conteneur Alpine Linux

Vous trouverez ci-dessous un exemple de Dockerfile qui peut être utilisé pour obtenir une installation fonctionnelle du AWS CLI dans un conteneur Alpine Linux comme alternative aux binaires prédéfinis pour Alpine. Lorsque vous utilisez cet exemple, remplacez-le *AWSCLI_VERSION* par le numéro de AWS CLI version souhaité :

```
FROM python:3.8-alpine AS builder

ENV AWSCLI_VERSION=2.10.1

RUN apk add --no-cache \
    curl \
    make \
    cmake \
    gcc \
    g++ \
    libc-dev \
    libffi-dev \
    openssl-dev \
    && curl https://awscli.amazonaws.com/awscli-${AWSCLI_VERSION}.tar.gz | tar -xz \
    && cd awscli-${AWSCLI_VERSION} \
    && ./configure --prefix=/opt/aws-cli/ --with-download-deps \
    && make \
    && make install

FROM python:3.8-alpine

RUN apk --no-cache add groff

COPY --from=builder /opt/aws-cli/ /opt/aws-cli/

ENTRYPOINT ["/opt/aws-cli/bin/aws"]
```

Cette image est créée puis AWS CLI invoquée à partir d'un conteneur similaire à celui créé sur Amazon Linux 2 :

```
$ docker build --tag awscli-alpine .
$ docker run --rm -it awscli-alpine --version
aws-cli/2.2.1 Python/3.8.11 Linux/5.10.25-linuxkit source-sandbox/x86_64.alpine.3
 prompt/off
```

La taille finale de cette image est inférieure à celle de l'image AWS CLI Docker officielle. Pour plus d'informations sur l'image officielle de Docker, consultezthe section called “Amazon ECR Public/ Docker”.

Résolution des erreurs AWS CLI d'installation et de désinstallation

Pour les étapes de résolution des erreurs d'installation, consultez the section called “Résoudre les erreurs” les étapes de dépannage courantes. Pour connaître les étapes de résolution des problèmes les plus pertinentesthe section called “Erreurs de commande non trouvées”, reportez-vous aux the section called “La commande « `aws --version` » renvoie une version différente de celle que vous avez installée” sections, etthe section called “La commande « `aws --version` » renvoie une version après la désinstallation du AWS CLI”.

Pour tout problème non traité dans les guides de dépannage, recherchez les problèmes avec l'`source-distribution`étiquette figurant dans le AWS CLI référentiel sur GitHub. Si aucun problème existant ne couvre vos erreurs, créez un nouveau problème pour bénéficier de l'aide des AWS CLI responsables.

Étapes suivantes

Après avoir installé le AWS CLI, vous devez effectuer unthe section called “Configuration”.

Exécution des images officielles Amazon ECR Public ou Docker pour AWS CLI

Cette rubrique décrit comment exécuter, contrôler les versions et configurer la AWS CLI version 2 sur Docker à l'aide de l'image officielle Amazon Elastic Container Registry Public (Amazon ECR Public) ou de Docker Hub. Pour de plus amples informations sur l'utilisation de Docker, veuillez consulter la Documentation de Docker.

Les images officielles offrent une isolation, une portabilité et une sécurité qui les soutiennent et les maintiennent AWS directement. Cela vous permet d'utiliser la AWS CLI version 2 dans un environnement basé sur des conteneurs sans avoir à gérer vous-même l'installation.

Rubriques

- Prérequis
- Choisir entre Amazon ECR Public et Docker Hub
- Exécutez les images de AWS CLI la version officielle 2
- Remarques sur les interfaces et la rétrocompatibilité des images officielles
- Utiliser des versions et des balises spécifiques
- Mise à jour vers la dernière image officielle
- Partage de fichiers hôtes, d'informations d'identification, de variables d'environnement et de configuration
- Raccourcissez la commande docker run

Prérequis

Docker doit être installé. Pour obtenir les instructions d'installation, consultez le site web Docker.

Pour vérifier votre installation de Docker, exécutez la commande suivante et vérifiez qu'il existe une sortie.

```
$ docker --version
Docker version 19.03.1
```

Choisir entre Amazon ECR Public et Docker Hub

Nous vous recommandons d'utiliser Amazon ECR Public via Docker Hub pour les AWS CLI images. Docker Hub impose des limites de débit plus strictes pour les consommateurs publics, ce qui peut entraîner des problèmes de régulation. En outre, Amazon ECR Public réplique les images dans plusieurs régions afin de garantir une disponibilité optimale et de gérer les problèmes de panne régionaux.

Pour plus d'informations sur la limitation du débit de Docker Hub, voir Comprendre la limitation du débit de Docker Hub sur le site Web de Docker.

Exécutez les images de AWS CLI la version officielle 2

La première fois que vous utilisez la `docker run` commande, la dernière image est téléchargée sur votre ordinateur. Chaque utilisation ultérieure de la commande `docker run` s'exécute à partir de votre copie locale.

Pour exécuter les images Docker AWS CLI version 2, utilisez la `docker run` commande.

Amazon ECR Public

L'image officielle Amazon ECR Public AWS CLI version 2 est hébergée sur Amazon ECR Public dans le `aws-cli/aws-cli` référentiel.

```
$ docker run --rm -it public.ecr.aws/aws-cli/aws-cli command
```

Docker Hub

L'image Docker officielle de la AWS CLI version 2 est hébergée sur Docker Hub dans le `amazon/aws-cli` référentiel.

```
$ docker run --rm -it amazon/aws-cli command
```

La commande fonctionne de la manière suivante :

- `docker run --rm -it` *`repository/name`*— L'équivalent de l'`aws`exécutable. Chaque fois que vous exécutez cette commande, Docker lance un conteneur contenant l'image que vous avez téléchargée et exécute votre `aws` commande. Par défaut, l'image utilise la dernière version de la AWS CLI version 2.

  Par exemple, pour appeler la commande `aws --version` dans Docker, vous exécutez ce qui suit.

  Amazon ECR Public

  ```
  $ docker run --rm -it public.ecr.aws/aws-cli/aws-cli --version
  aws-cli/2.27.41 Python/3.7.3 Linux/4.9.184-linuxkit botocore/2.4.5dev10
  ```

  Docker Hub

  ```
  $ docker run --rm -it amazon/aws-cli --version
  ```

```
aws-cli/2.27.41 Python/3.7.3 Linux/4.9.184-linuxkit botocore/2.4.5dev10
```

- `--rm`— Spécifie de nettoyer le conteneur après la fin de la commande.
- `-it`— Spécifie l'ouverture d'un pseudo-TTY avec`stdin`. Cela vous permet de fournir des informations à la AWS CLI version 2 lorsqu'elle s'exécute dans un conteneur, par exemple à l'aide des `aws help` commandes `aws configure` et. Lorsque vous choisissez d'omettre ou non`-it`, tenez compte des points suivants :
  - Si vous exécutez des scripts, ce n'`-it`est pas nécessaire.
  - Si vous rencontrez des erreurs avec vos scripts, le fait de ne pas participer `-it` à votre appel Docker peut résoudre le problème.
  - Si vous essayez de rediriger la sortie, `-it` cela peut provoquer des erreurs et le fait `-it` de ne pas participer à votre appel Docker peut résoudre ce problème. Si vous souhaitez conserver l'`-it`indicateur, mais que vous souhaitez tout de même rediriger la sortie, la désactivation du pager côté client AWS CLI utilisé par défaut devrait résoudre le problème.

Pour de plus amples informations sur la commande `docker run`, veuillez consulter le guide de référence Docker.

Remarques sur les interfaces et la rétrocompatibilité des images officielles

- Le seul outil pris en charge sur l'image est le AWS CLI. Seul le `aws` fichier exécutable doit être exécuté directement. Par exemple, même si `less` et `groff` sont explicitement installés sur l'image, ils ne doivent pas être exécutés directement en dehors d'une AWS CLI commande.
- Le répertoire `/aws` de travail est contrôlé par l'utilisateur. L'image n'écrira pas dans ce répertoire, sauf indication contraire de l'utilisateur lors de l'exécution d'une AWS CLI commande.
- Il n'y a aucune garantie de rétrocompatibilité en s'appuyant sur la dernière balise. Pour garantir la rétrocompatibilité, vous devez épingler une `<major.minor.patch>` balise spécifique car ces balises sont immuables ; elles ne seront reportées qu'une seule fois.

Utiliser des versions et des balises spécifiques

L'image officielle de la AWS CLI version 2 comporte plusieurs versions que vous pouvez utiliser, à commencer par la version`2.0.6`. Pour exécuter une version spécifique de la AWS CLI version 2, ajoutez la balise appropriée à votre `docker run` commande. La première fois que vous utilisez la `docker run` commande avec une balise, la dernière image associée à cette balise est téléchargée

sur votre ordinateur. Chaque utilisation ultérieure de la commande `docker run` avec cette balise s'exécute à partir de votre copie locale.

Vous pouvez utiliser deux types de balises :

- `latest`— Définit la dernière version de la AWS CLI version 2 pour l'image. Nous vous recommandons d'utiliser le `latest` tag lorsque vous souhaitez disposer de la dernière version de la AWS CLI version 2. Cependant, il n'y a aucune garantie de compatibilité descendante lorsque vous utilisez cette balise. La balise `latest` est utilisée par défaut dans la commande `docker run`. Pour utiliser explicitement la balise `latest`, ajoutez la balise au nom de l'image du conteneur.

  Amazon ECR Public

  ```
  $ docker run --rm -it public.ecr.aws/aws-cli/aws-cli:latest command
  ```

  Docker Hub

  ```
  $ docker run --rm -it amazon/aws-cli:latest command
  ```

- `<major.minor.patch>`— Définit une version spécifique de la AWS CLI version 2 pour l'image. Si vous envisagez d'utiliser une image officielle en production, nous vous recommandons d'utiliser une version spécifique de la AWS CLI version 2 pour garantir la rétrocompatibilité. Par exemple, pour exécuter la version`2.0.6`, ajoutez la version au nom de l'image du conteneur.

  Amazon ECR Public

  ```
  $ docker run --rm -it public.ecr.aws/aws-cli/aws-cli:2.0.6 command
  ```

  Docker Hub

  ```
  $ docker run --rm -it amazon/aws-cli:2.0.6 command
  ```

Mise à jour vers la dernière image officielle

Comme la dernière image n'est téléchargée sur votre ordinateur que la première fois que vous utilisez la `docker run` commande, vous devez extraire manuellement une image mise à jour. Pour mettre à jour manuellement vers la dernière version, nous vous recommandons d'extraire

l'image balisée `latest`. L'extraction de l'image permet de télécharger la dernière version sur votre ordinateur.

Amazon ECR Public

```
$ docker pull public.ecr.aws/aws-cli/aws-cli:latest
```

Docker Hub

```
$ docker pull amazon/aws-cli:latest
```

Partage de fichiers hôtes, d'informations d'identification, de variables d'environnement et de configuration

La AWS CLI version 2 étant exécutée dans un conteneur, la CLI ne peut pas accéder par défaut au système de fichiers hôte, qui inclut la configuration et les informations d'identification. Pour partager le système de fichiers hôte, les informations d'identification et la configuration avec le conteneur, montez le répertoire `~/.aws` du système hôte sur le conteneur à l'adresse `/root/.aws` avec l'indicateur `-v` défini sur la commande `docker run`. Cela permet à la AWS CLI version 2 exécutée dans le conteneur de localiser les informations du fichier hôte.

Amazon ECR Public

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws public.ecr.aws/aws-cli/aws-cli command
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws public.ecr.aws/aws-cli/aws-
cli command
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws  public.ecr.aws/aws-
cli/aws-cli command
```

Docker Hub

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws amazon/aws-cli command
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws amazon/aws-cli command
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws  amazon/aws-cli command
```

Pour de plus amples informations sur l'indicateur `-v` et le montage, veuillez consulter le guide de référence Docker.

> **Note**
>
> Pour plus d'informations sur `credentials` les fichiers `config` et, voirthe section called “Paramètres de configuration”.

Exemple 1 : Fourniture d'informations d'identification et configuration

Dans cet exemple, nous fournissons les informations d'identification et la configuration de l'hôte lors de l'exécution de la `s3 ls` commande pour répertorier vos buckets dans Amazon Simple Storage Service (Amazon S3). Les exemples ci-dessous utilisent l'emplacement par défaut pour les AWS CLI informations d'identification et les fichiers de configuration. Pour utiliser un autre emplacement, modifiez le chemin du fichier.

Amazon ECR Public

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws public.ecr.aws/aws-cli/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws public.ecr.aws/aws-cli/aws-
cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws public.ecr.aws/aws-cli/
aws-cli s3 ls
```

Docker Hub

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws amazon/aws-cli s3 ls
```

Vous pouvez appeler des variables d'environnement spécifiques à un système à l'aide de l'-eindicateur. Pour utiliser une variable d'environnement, appelez-la par son nom.

Amazon ECR Public

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e ENVVAR_NAME public.ecr.aws/aws-cli/
aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws -e ENVVAR_NAME
 public.ecr.aws/aws-cli/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -e ENVVAR_NAME
 public.ecr.aws/aws-cli/aws-cli s3 ls
```

Docker Hub

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e ENVVAR_NAME amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws -e ENVVAR_NAME amazon/aws-cli
 s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -e ENVVAR_NAME amazon/
aws-cli s3 ls
```

Exemple 2 : téléchargement d'un fichier Amazon S3 sur votre système hôte

Pour certaines commandes de AWS CLI la version 2, vous pouvez lire des fichiers depuis le système hôte dans le conteneur ou écrire des fichiers depuis le conteneur vers le système hôte.

Dans cet exemple, nous téléchargeons l'objet S3 `s3://aws-cli-docker-demo/hello` sur votre système de fichiers local en installant le répertoire de travail actuel dans le répertoire `/aws` du conteneur. En téléchargeant l'objet `hello` dans le répertoire `/aws` du conteneur, le fichier est également enregistré dans le répertoire de travail actuel du système hôte.

Amazon ECR Public

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws public.ecr.aws/aws-cli/
aws-cli s3 cp s3://aws-cli-docker-demo/hello .
download: s3://aws-cli-docker-demo/hello to ./hello
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws -v %cd%:/aws public.ecr.aws/
aws-cli/aws-cli s3 cp s3://aws-cli-docker-demo/hello .
download: s3://aws-cli-docker-demo/hello to ./hello
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -v $pwd\aws:/aws
 public.ecr.aws/aws-cli/aws-cli s3 cp s3://aws-cli-docker-demo/hello .
```

Docker Hub

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws amazon/aws-cli s3 cp s3://
aws-cli-docker-demo/hello .
download: s3://aws-cli-docker-demo/hello to ./hello
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws -v %cd%:/aws amazon/aws-cli
 s3 cp s3://aws-cli-docker-demo/hello .
download: s3://aws-cli-docker-demo/hello to ./hello
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -v $pwd\aws:/aws
 amazon/aws-cli s3 cp s3://aws-cli-docker-demo/hello .
```

Pour confirmer que le fichier téléchargé existe dans le système de fichiers local, exécutez la procédure suivante.

Linux et macOS

```
$ cat hello
Hello from Docker!
```

Windows PowerShell

```
$ type hello
Hello from Docker!
```

Exemple 3 : Utilisation de votre variable d' AWS_PROFILEenvironnement

Vous pouvez appeler des variables d'environnement spécifiques à un système à l'aide de l'`-e`indicateur. Appelez chaque variable d'environnement que vous souhaitez utiliser. Dans cet exemple, nous fournissons les informations d'identification de l'hôte, la configuration et la variable d'*`AWS_PROFILE`*environnement lors de l'exécution de la `s3 ls` commande pour répertorier vos buckets dans Amazon Simple Storage Service (Amazon S3).

Amazon ECR Public

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e AWS_PROFILE public.ecr.aws/aws-cli/
aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws -e AWS_PROFILE
 public.ecr.aws/aws-cli/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -e AWS_PROFILE
 public.ecr.aws/aws-cli/aws-cli s3 ls
```

Docker Hub

Linux et macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e AWS_PROFILE amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Invite de commande Windows

```
$ docker run --rm -it -v %userprofile%\.aws:/root/.aws -e AWS_PROFILE amazon/aws-cli
 s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -e AWS_PROFILE amazon/
aws-cli s3 ls
```

Raccourcissez la commande docker run

Pour raccourcir la `docker run` commande, nous vous suggérons d'utiliser la capacité de votre système d'exploitation pour créer un `symbolic link`(lien symbolique) ou `alias`sous Linux et macOS, ou `doskey`sous Windows. Pour définir l'alias `aws`, vous pouvez exécuter l'une des commandes suivantes.

- Pour un accès de base aux commandes `aws`, exécutez la procédure suivante.

  Amazon ECR Public

  Linux et macOS

  ```
  $ alias aws='docker run --rm -it public.ecr.aws/aws-cli/aws-cli'
  ```

  Invite de commande Windows

  ```
  C:\> doskey aws=docker run --rm -it public.ecr.aws/aws-cli/aws-cli $*
  ```

  Windows PowerShell

  ```
  C:\> Function AWSCLI {docker run --rm -it public.ecr.aws/aws-cli/aws-cli $args}
  Set-Alias -Name aws -Value AWSCLI
  ```

Docker Hub

Linux et macOS

```
$ alias aws='docker run --rm -it amazon/aws-cli'
```

Invite de commande Windows

```
C:\> doskey aws=docker run --rm -it amazon/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it amazon/aws-cli $args}
Set-Alias -Name aws -Value AWSCLI
```

- Pour accéder au système de fichiers hôte et aux paramètres de configuration lors de l'utilisation des commandes `aws`, exécutez la procédure suivante.

Amazon ECR Public

Linux et macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws
 public.ecr.aws/aws-cli/aws-cli'
```

Invite de commande Windows

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\.aws:/root/.aws -v %cd%:/aws
 public.ecr.aws/aws-cli/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\.aws:/root/.aws -v
 $pwd\aws:/aws public.ecr.aws/aws-cli/aws-cli $args}
Set-Alias -Name aws -Value AWSCLI
```

Docker Hub

Linux et macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws amazon/aws-
cli'
```

Invite de commande Windows

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\.aws:/root/.aws -v %cd%:/aws
 amazon/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\.aws:/root/.aws -v
 $pwd\aws:/aws amazon/aws-cli $args}
Set-Alias -Name aws -Value AWSCLI
```

- Pour affecter une version spécifique à utiliser dans votre alias aws, ajoutez votre balise de version.

  Amazon ECR Public

  Linux et macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws
 public.ecr.aws/aws-cli/aws-cli:2.0.6'
```

  Invite de commande Windows

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\.aws:/root/.aws -v %cd%:/aws
 public.ecr.aws/aws-cli/aws-cli:2.0.6 $*
```

  Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\.aws:/root/.aws -v
 $pwd\aws:/aws public.ecr.aws/aws-cli/aws-cli:2.0.6 $args}
Set-Alias -Name aws -Value AWSCLI
```

  Docker Hub

  Linux et macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws amazon/aws-
cli:2.0.6'
```

Invite de commande Windows

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\.aws:/root/.aws -v %cd%:/aws
 amazon/aws-cli:2.0.6 $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\.aws:/root/.aws -v
 $pwd\aws:/aws amazon/aws-cli:2.0.6 $args}
Set-Alias -Name aws -Value AWSCLI
```

Après avoir défini votre alias, vous pouvez exécuter la AWS CLI version 2 depuis un conteneur comme si elle était installée sur votre système hôte.

```
$ aws --version
aws-cli/2.27.41 Python/3.7.3 Linux/4.9.184-linuxkit botocore/2.4.5dev10
```

Configuration du AWS CLI

Cette rubrique explique comment configurer rapidement les paramètres de base avec lesquels le AWS Command Line Interface (AWS CLI) interagit AWS. Il s'agit notamment de vos informations de sécurité, du format de sortie par défaut et de la AWS région par défaut.

Rubriques

- Rassemblez vos informations d'identification pour un accès programmatique
- Configuration d'une nouvelle configuration et de nouvelles informations d'identification
- Utilisation des fichiers de configuration et d'identification existants

Rassemblez vos informations d'identification pour un accès programmatique

Vous aurez besoin d'un accès programmatique si vous souhaitez interagir avec AWS l'extérieur du AWS Management Console. Pour les instructions d'authentification et d'identification, choisissez l'une des options suivantes :

Type d’authentification	Objectif	Instructions
Identifiants à court terme des utilisateurs du personnel d'IAM Identity Center	(Recommandé) Utilisez des informations d'identification à court terme pour un utilisateur du personnel d'IAM Identity Center. La meilleure pratique en matière de sécurité consiste AWS Organizations à utiliser IAM Identity Center. Il combine des informations d'identification à court terme avec un annuaire d'utilisateurs, tel que le répertoire intégré IAM Identity Center ou Active Directory.	the section called “Authentification IAM Identity Center”
Informations d'identification à court terme de l'utilisateur IAM	Utilisez les informations d'identification à court terme des utilisateurs IAM, qui sont plus sécurisées que les informations d'identification à long terme. Si vos informations d'identification sont compromises, elles peuvent être utilisées pendant une durée limitée avant leur expiration.	the section called “Identifiants à court terme”

Type d’authentification	Objectif	Instructions
Utilisateurs d'IAM ou IAM Identity Center sur une instance Amazon EC2 .	Utilisez les métadonnées de l' EC2 instance Amazon pour demander des informations d'identification temporaires à l'aide du rôle attribué à l' EC2 instance Amazon.	the section called “ EC2 Métadonnées Amazon”
Assumer des rôles pour les autorisations	Associez une autre méthode d'identification et assumez un rôle pour un accès temporair e auquel Services AWS votre utilisateur n'a peut-être pas accès.	the section called “Rôles IAM”
Informations d'identification à long terme de l'utilisateur IAM	(Non recommandé) Utilisez des informations d'identif ication à long terme, qui n'ont pas d'expiration.	the section called “Utilisateurs IAM”
Stockage externe des utilisate urs du personnel d'IAM ou d'IAM Identity Center	(Non recommandé) Associez une autre méthode d'identif ication mais stockez les valeurs d'identification dans un emplacement extérieur à. AWS CLI La sécurité de cette méthode dépend de l'emplace ment externe dans lequel les informations d'identification sont stockées.	the section called “External Credentials”

Configuration d'une nouvelle configuration et de nouvelles informations d'identification

AWS CLI Stocke vos informations de configuration et d'identification dans un profil (un ensemble de paramètres) dans les `config` fichiers `credentials` et.

Il existe principalement deux méthodes pour effectuer une configuration rapide :

- Configuration à l'aide de AWS CLI commandes
- Modification manuelle des informations d'identification et des fichiers de configuration

Les exemples suivants utilisent des exemples de valeurs pour chacune des méthodes d'authentification. Remplacez les valeurs d'échantillon par les vôtres.

Configuration à l'aide de AWS CLI commandes

Pour une utilisation générale, les `aws configure sso` commandes `aws configure` ou de votre terminal préféré constituent le moyen le plus rapide de configurer votre AWS CLI installation. En fonction de la méthode d'identification que vous préférez, vous êtes AWS CLI invité à saisir les informations pertinentes. Par défaut, les informations de ce profil sont utilisées lorsque vous exécutez une AWS CLI commande qui ne spécifie pas explicitement le profil à utiliser.

Pour plus d'informations sur les `config` fichiers `credentials` et, consultezParamètres des fichiers de configuration et d'identification dans le AWS CLI.

IAM Identity Center (SSO)

Cet exemple concerne AWS IAM Identity Center l'utilisation de l'`aws configure sso`assistant. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

Attempting to automatically open the SSO authorization page in your default browser.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
```

```
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

CLI default client Region [None]: us-west-2
CLI default output format [None]: json
CLI profile name [123456789011_ReadOnly]: user1
```

IAM Identity Center (Legacy SSO)

Cet exemple concerne l'ancienne méthode d' AWS IAM Identity Center utilisation de l'`aws configure sso` assistant. Pour utiliser l'ancienne authentification unique, laissez le nom de session vide. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

```
$ aws configure sso
SSO session name (Recommended):
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

CLI default client Region [None]: us-west-2
CLI default output format [None]: json
```

```
CLI profile name [123456789011_ReadOnly]: user1
```

Short-term credentials

Cet exemple concerne les informations d'identification à court terme de AWS Identity and Access Management. L'assistant de configuration aws est utilisé pour définir les valeurs initiales, puis la `aws configure set` commande attribue la dernière valeur requise. Pour de plus amples informations, veuillez consulter the section called “Identifiants à court terme”.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
$ aws configure set
 aws_session_token fcZib3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

IAM role

Cet exemple permet d'assumer un rôle IAM. Les profils qui utilisent des rôles IAM extraient les informations d'identification d'un autre profil, puis appliquent les autorisations des rôles IAM. Dans les exemples suivants, `default` est le profil source des informations d'identification et `user1` emprunte les mêmes informations d'identification puis assume un nouveau rôle. Il n'y a pas d'assistant pour ce processus, donc chaque valeur est définie à l'aide de la `aws configure set` commande. Pour de plus amples informations, veuillez consulter the section called “Rôles IAM”.

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
$ aws configure set source_profile default
$ aws configure set role_session_name session_user1
$ aws configure set region us-west-2
$ aws configure set output json
```

Amazon EC2 instance metadata credentials

Cet exemple concerne les informations d'identification obtenues à partir des métadonnées de l' EC2 instance Amazon d'hébergement. Il n'y a pas d'assistant pour ce processus, donc chaque valeur est définie à l'aide de la `aws configure set` commande. Pour de plus amples informations, veuillez consulter the section called “ EC2 Métadonnées Amazon”.

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
```

```
$ aws configure set credential_source Ec2InstanceMetadata
$ aws configure set region us-west-2
$ aws configure set output json
```

Long-term credentials

> **Warning**
>
> Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que AWS IAM Identity Center.

Cet exemple concerne les informations d'identification à long terme de AWS Identity and Access Management. Pour de plus amples informations, veuillez consulter the section called "Utilisateurs IAM".

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Pour des informations plus détaillées sur les méthodes d'authentification et d'identification, voirAuthentification et identifiants d'accès.

Modification manuelle des informations d'identification et des fichiers de configuration

Lorsque vous copiez et collez des informations, nous vous suggérons de modifier manuellement le `credentials` fichier `config` et. En fonction de la méthode d'identification que vous préférez, les fichiers sont configurés différemment.

Les fichiers sont stockés dans votre répertoire personnel, sous le `.aws` dossier. L'emplacement du répertoire de base varie selon le système d'exploitation, mais il est référencé à l'aide de variables d'environnement `%UserProfile%` sous Windows et `$HOME` ou ~ (tilde) sur les systèmes Unix. Pour plus d'informations sur l'emplacement de stockage de ces paramètres, consultezthe section called "Où les paramètres de configuration sont-ils stockés ?".

Les exemples suivants montrent un `default` profil et un profil nommés `user1` et utilisent des exemples de valeurs. Remplacez les valeurs d'échantillon par les vôtres. Pour plus d'informations sur les `config` fichiers `credentials` et, consultezParamètres des fichiers de configuration et d'identification dans le AWS CLI.

IAM Identity Center (SSO)

Cet exemple est pour AWS IAM Identity Center. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

Fichier d'informations d'identification

Le `credentials` fichier n'est pas utilisé pour cette méthode d'authentification.

Fichier de configuration

```
[default]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text

[profile user1]
sso_session = my-sso
sso_account_id = 444455556666
sso_role_name = readOnly
region = us-east-1
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

IAM Identity Center (Legacy SSO)

Cet exemple concerne l'ancienne méthode de AWS IAM Identity Center. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

Fichier d'informations d'identification

Le `credentials` fichier n'est pas utilisé pour cette méthode d'authentification.

Fichier de configuration

```
[default]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text

[profile user1]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 444455556666
sso_role_name = readOnly
region = us-east-1
output = json
```

Short-term credentials

Cet exemple concerne les informations d'identification à court terme de AWS Identity and Access Management. Pour de plus amples informations, veuillez consulter the section called “Identifiants à court terme”.

Fichier d'informations d'identification

```
[default]
aws_access_key_id=ASIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token
 = IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE

[user1]
aws_access_key_id=ASIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
aws_session_token
 = fcZib3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

Fichier de configuration

```
[default]
region=us-west-2
output=json
```

```
[profile user1]
region=us-east-1
output=text
```

IAM role

Cet exemple permet d'assumer un rôle IAM. Les profils qui utilisent des rôles IAM extraient les informations d'identification d'un autre profil, puis appliquent les autorisations des rôles IAM. Dans les exemples suivants, `default` est le profil source des informations d'identification et `user1` emprunte les mêmes informations d'identification puis assume un nouveau rôle. Pour de plus amples informations, veuillez consulter the section called “Rôles IAM”.

Fichier d'informations d'identification

Le `credentials` fichier dépend de l'authentification utilisée par votre profil source. Dans l'exemple suivant, le profil source utilise des informations d'identification à court terme.

```
[default]
aws_access_key_id=ASIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token
 = IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

Fichier de configuration

```
[default]
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
source_profile=default
role_session_name=session_user1
region=us-east-1
output=text
```

Amazon EC2 instance metadata credentials

Cet exemple concerne les informations d'identification obtenues à partir des métadonnées de l' EC2 instance Amazon d'hébergement. Pour de plus amples informations, veuillez consulter the section called “ EC2 Métadonnées Amazon”.

Fichier d'informations d'identification

Le `credentials` fichier n'est pas utilisé pour cette méthode d'authentification.

Fichier de configuration

```
[default]
role_arn=arn:aws:iam::123456789012:role/defaultrole
credential_source=Ec2InstanceMetadata
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
credential_source=Ec2InstanceMetadata
region=us-east-1
output=text
```

Long-term credentials

> **Warning**
>
> Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que AWS IAM Identity Center.

Cet exemple concerne les informations d'identification à long terme de AWS Identity and Access Management. Pour de plus amples informations, veuillez consulter the section called “Utilisateurs IAM”.

Fichier d'informations d'identification

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

[user1]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

Fichier de configuration

```
[default]
region=us-west-2
output=json

[profile user1]
region=us-east-1
output=text
```

Pour des informations plus détaillées sur les méthodes d'authentification et d'identification, voirAuthentification et identifiants d'accès.

Utilisation des fichiers de configuration et d'identification existants

Si vous disposez de fichiers de configuration et d'informations d'identification existants, ceux-ci peuvent être utilisés pour le AWS CLI.

Pour utiliser les `credentials` fichiers `config` et, déplacez-les vers le dossier nommé `.aws` dans votre répertoire personnel. L'emplacement du répertoire de base varie selon le système d'exploitation, mais il est référencé à l'aide de variables d'environnement `%UserProfile%` sous Windows et `$HOME` ou ~ (tilde) sur les systèmes Unix.

Vous pouvez spécifier un emplacement autre que celui par défaut pour les `credentials` fichiers `config` et en définissant les variables d'`AWS_SHARED_CREDENTIALS_FILE`environnement `AWS_CONFIG_FILE` et sur un autre chemin local. Consultez Configuration des variables d'environnement pour AWS CLI pour plus de détails.

Pour des informations plus détaillées sur les fichiers de configuration et d'identification, consultezthe section called “Paramètres de configuration”.

Configuration des paramètres pour AWS CLI

Cette section explique comment configurer les paramètres avec lesquels le AWS Command Line Interface (AWS CLI) interagit AWS. Tel est le cas des éléments suivants :

- Les informations d'identification identifient la personne qui appelle l'API. Les informations d'accès sont utilisées pour chiffrer la demande adressée aux AWS serveurs afin de confirmer votre identité et de récupérer les politiques d'autorisation associées. Ces autorisations déterminent les actions que vous pouvez effectuer. Pour plus d'informations sur la configuration de vos informations d'identification, consultezAuthentification et identifiants d'accès.
- Autres détails de configuration indiquant AWS CLI comment traiter les demandes, tels que le format de sortie par défaut et la AWS région par défaut.

> **Note**
>
> AWS exige que toutes les demandes entrantes soient signées de manière cryptographique. Il le AWS CLI fait pour toi. La « signature » inclut un date/time cachet. Par conséquent, vous devez veiller à ce que la date et l'heure de votre ordinateur soient correctement définies. Si ce n'est pas le date/time cas et que le contenu de la signature est trop éloigné de date/time celui reconnu par le AWS service, AWS rejette la demande.

Priorité de la configuration et des informations d'identification

Les informations d'identification et les paramètres de configuration se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. Certains emplacements l'emportent sur d'autres. Les AWS CLI informations d'identification et les paramètres de configuration sont prioritaires dans l'ordre suivant :

1. Options de ligne de commande : remplace les paramètres de tout autre emplacement, tels que les `--profile` paramètres `--region``--output`, et.
2. Variables d'environnement : vous pouvez stocker des valeurs dans les variables d'environnement de votre système.
3. Assumer un rôle — Assumez les autorisations d'un rôle IAM par le biais de la configuration ou de la commande.

4. Assumer un rôle avec une identité Web : assumez les autorisations d'un rôle IAM en utilisant l'identité Web par le biais de la configuration ou de la commande.
5. AWS IAM Identity Center— Les paramètres de configuration d'IAM Identity Center stockés dans le `config` fichier sont mis à jour lorsque vous exécutez la `aws configure sso` commande. Les informations d'identification sont ensuite authentifiées lorsque vous exécutez la `aws sso login` commande. Le `config` fichier se trouve sous `~/.aws/config` Linux ou macOS, ou `C:\Users\`*`USERNAME`*`\.aws\config` sous Windows.
6. Fichier d'informations d'identification : le `config` fichier `credentials` et est mis à jour lorsque vous exécutez la commande`aws configure`. Le `credentials` fichier se trouve sous `~/.aws/credentials` Linux ou macOS, ou `C:\Users\`*`USERNAME`*`\.aws\credentials` sous Windows.
7. Processus personnalisé — Obtenez vos informations d'identification auprès d'une source externe.
8. Fichier de configuration — Le `config` fichier `credentials` et est mis à jour lorsque vous exécutez la commande`aws configure`. Le `config` fichier se trouve sous `~/.aws/config` Linux ou macOS, ou `C:\Users\`*`USERNAME`*`\.aws\config` sous Windows.
9. Informations d'identification du conteneur : vous pouvez associer un rôle IAM à chacune de vos définitions de tâches Amazon Elastic Container Service (Amazon ECS). Les informations d'identification temporaires pour ce rôle sont ensuite disponibles pour les conteneurs de cette tâche. Pour plus d'informations, consultez Rôles IAM pour les tâches dans le Guide du développeur Amazon Elastic Container Service.
10. Informations d'identification du profil d' EC2instance Amazon : vous pouvez associer un rôle IAM à chacune de vos instances Amazon Elastic Compute Cloud (Amazon EC2). Les informations d'identification temporaires pour ce rôle sont ensuite disponibles pour l'exécution du code dans l'instance. Les informations d'identification sont fournies via le service de EC2 métadonnées Amazon. Pour plus d'informations, consultez les sections Rôles IAM pour Amazon EC2 dans le guide de l' EC2 utilisateur Amazon et Utilisation des profils d'instance dans le guide de l'utilisateur IAM.

Sujets supplémentaires dans cette section

- the section called “Paramètres de configuration”
- the section called “Variables d'environnement”
- the section called “Options de ligne de commande”
- the section called “ Saisie automatique de commandes”
- the section called “Nouvelle tentative”

- the section called “Proxys HTTP”

Paramètres des fichiers de configuration et d'identification dans le AWS CLI

Vous pouvez enregistrer vos paramètres de configuration utilisés fréquemment et vos informations d'identification dans les fichiers qui sont gérés par l' AWS CLI.

Les fichiers sont divisés en `profiles`. Par défaut, AWS CLI utilise les paramètres du profil nommé`default`. Pour utiliser d'autres paramètres, vous pouvez créer d'autres profils et y faire référence.

Vous pouvez également remplacer un paramètre spécifique en définissant l'une des variables d'environnement prises en charge ou un paramètre de ligne de commande. Pour plus d'informations sur la priorité des paramètres de configuration, veuillez consulter Configuration des paramètres pour AWS CLI.

> **Note**
>
> Pour plus d'informations sur la configuration de vos informations d'identification, consultezAuthentification et identifiants d'accès.

Rubriques

- Format des fichiers de configuration et d'identification
- Où les paramètres de configuration sont-ils stockés ?
- Utilisation de profils nommés
- Définir et afficher les paramètres de configuration à l'aide de commandes
- Exemples de commandes de définition d'une nouvelle configuration et d'informations d'identification
- Paramètres de fichier config pris en charge

Format des fichiers de configuration et d'identification

Les `credentials` fichiers `config` et sont organisés en sections. Les sections incluent les profils, les sessions SSO et les services. Une section est un ensemble nommé de paramètres qui continue

jusqu'à ce qu'une autre ligne de définition de section soit rencontrée. Plusieurs profils et sections peuvent être enregistrés dans les `credentials` fichiers `config` et.

Ces fichiers sont des fichiers en texte brut qui utilisent le format suivant :

- Les noms des sections sont placés entre crochets [] tels que `[default]`[`profile` *`user1`*], et`[sso-session]`.
- Toutes les entrées d'une section prennent la forme générale d`esetting_name=value`.
- Les lignes peuvent être commentées en commençant par un caractère dièse (#).

Les credentials fichiers config et contiennent les types de sections suivants :

- Type de section : profile
- Type de section : sso-session
- Type de section : services

Type de section : **`profile`**

Selon le fichier, les noms des sections de profil utilisent le format suivant :

- Fichier de configuration : `[default]` `[profile` *`user1`*`]`
- Fichier d'informations d'identification : `[default]` [*`user1`*`]`

  N’utilisez pas le mot `profile` lors de la création d’une entrée dans le fichier `credentials`.

Chaque profil peut spécifier des informations d'identification différentes et peut également spécifier des AWS régions et des formats de sortie différents. Lorsque vous nommez le profil dans un `config` fichier, incluez le préfixe `profile` « », mais ne l'incluez pas dans le `credentials` fichier.

Les exemples suivants montrent un `config` fichier `credentials` et avec deux profils, une région et une sortie spécifiés. Le premier [par défaut] est utilisé lorsque vous exécutez une AWS CLI commande sans qu'aucun profil ne soit spécifié. Le second est utilisé lorsque vous exécutez une AWS CLI commande avec le `--profile user1` paramètre.

IAM Identity Center (SSO)

> Cet exemple est pour AWS IAM Identity Center. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

Fichier d'informations d'identification

Le `credentials` fichier n'est pas utilisé pour cette méthode d'authentification.

Fichier de configuration

```
[default]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text

[profile user1]
sso_session = my-sso
sso_account_id = 444455556666
sso_role_name = readOnly
region = us-east-1
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

IAM Identity Center (Legacy SSO)

Cet exemple concerne l'ancienne méthode de AWS IAM Identity Center. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

Fichier d'informations d'identification

Le `credentials` fichier n'est pas utilisé pour cette méthode d'authentification.

Fichier de configuration

```
[default]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text
```

```
[profile user1]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 444455556666
sso_role_name = readOnly
region = us-east-1
output = json
```

Short-term credentials

Cet exemple concerne les informations d'identification à court terme de AWS Identity and Access Management. Pour de plus amples informations, veuillez consulter the section called “Identifiants à court terme”.

Fichier d'informations d'identification

```
[default]
aws_access_key_id=ASIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token
 = IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE

[user1]
aws_access_key_id=ASIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
aws_session_token
 = fcZib3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

Fichier de configuration

```
[default]
region=us-west-2
output=json

[profile user1]
region=us-east-1
output=text
```

IAM role

Cet exemple permet d'assumer un rôle IAM. Les profils qui utilisent des rôles IAM extraient les informations d'identification d'un autre profil, puis appliquent les autorisations des rôles IAM. Dans

les exemples suivants, `default` est le profil source des informations d'identification et `user1` emprunte les mêmes informations d'identification puis assume un nouveau rôle. Pour de plus amples informations, veuillez consulter the section called “Rôles IAM”.

Fichier d'informations d'identification

Le `credentials` fichier dépend de l'authentification utilisée par votre profil source. Dans l'exemple suivant, le profil source utilise des informations d'identification à court terme.

```
[default]
aws_access_key_id=ASIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token
 = IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

Fichier de configuration

```
[default]
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
source_profile=default
role_session_name=session_user1
region=us-east-1
output=text
```

Amazon EC2 instance metadata credentials

Cet exemple concerne les informations d'identification obtenues à partir des métadonnées de l' EC2 instance Amazon d'hébergement. Pour de plus amples informations, veuillez consulter the section called “ EC2 Métadonnées Amazon”.

Fichier d'informations d'identification

Le `credentials` fichier n'est pas utilisé pour cette méthode d'authentification.

Fichier de configuration

```
[default]
```

```
role_arn=arn:aws:iam::123456789012:role/defaultrole
credential_source=Ec2InstanceMetadata
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
credential_source=Ec2InstanceMetadata
region=us-east-1
output=text
```

Long-term credentials

> **Warning**
>
> Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que AWS IAM Identity Center.

Cet exemple concerne les informations d'identification à long terme de AWS Identity and Access Management. Pour de plus amples informations, veuillez consulter the section called "Utilisateurs IAM".

Fichier d'informations d'identification

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

[user1]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

Fichier de configuration

```
[default]
region=us-west-2
output=json
```

```
[profile user1]
region=us-east-1
output=text
```

Pour plus d'informations et des méthodes d'autorisation et d'identification supplémentaires, voirthe section called “Utilisateurs IAM”.

Type de section : **`sso-session`**

La `sso-session` section du `config` fichier est utilisée pour regrouper les variables de configuration permettant d'acquérir des jetons d'accès SSO, qui peuvent ensuite être utilisés pour acquérir des AWS informations d'identification. Les paramètres suivants sont utilisés :

- (Obligatoire) `sso_start_url`
- (Obligatoire) `sso_region`
- `sso_account_id`
- `sso_role_name`
- `sso_registration_scopes`

Vous définissez une `sso-session` section et vous l'associez à un profil. `sso_region`et `sso_start_url` doit être défini dans la `sso-session` section. Généralement, `sso_account_id` et `sso_role_name` doit être défini dans la `profile` section afin que le SDK puisse demander des informations d'identification SSO.

L'exemple suivant configure le SDK pour demander des informations d'identification SSO et prend en charge l'actualisation automatique des jetons :

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

Cela permet également de réutiliser les `sso-session` configurations sur plusieurs profils :

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

Cependant, ils `sso_account_id` `sso_role_name` ne sont pas obligatoires pour tous les scénarios de configuration de jetons SSO. Si votre application utilise uniquement des AWS services qui prennent en charge l'authentification par porteur, les informations d' AWS identification traditionnelles ne sont pas nécessaires. L'authentification du porteur est un schéma d'authentification HTTP qui utilise des jetons de sécurité appelés jetons porteurs. Dans ce scénario, `sso_account_id` et `sso_role_name` ne sont pas obligatoires. Consultez le guide individuel de votre AWS service pour déterminer s'il prend en charge l'autorisation par jeton au porteur.

De plus, les étendues d'enregistrement peuvent être configurées dans le cadre d'un`sso-session`. Scope est un mécanisme de la OAuth version 2.0 qui limite l'accès d'une application au compte d'un utilisateur. Une application peut demander une ou plusieurs étendues, et le jeton d'accès délivré à l'application sera limité aux étendues accordées. Ces étendues définissent les autorisations demandées pour être autorisées pour le client OIDC enregistré et les jetons d'accès récupérés par le client. L'exemple suivant fournit un accès `sso_registration_scopes` pour répertorier les comptes/rôles :

```
[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Le jeton d'authentification est mis en cache sur le disque sous le `~/.aws/sso/cache` répertoire avec un nom de fichier basé sur le nom de session.

Pour plus d'informations sur ce type de configuration, consultezthe section called “Authentification IAM Identity Center”.

Type de section : **services**

La `services` section est un groupe de paramètres qui configure les points de terminaison personnalisés pour les Service AWS demandes. Un profil est ensuite lié à une `services` section.

```
[profile dev]
services = my-services
```

La `services` section est séparée en sous-sections par des `<SERVICE> =` lignes, où se `<SERVICE>` trouve la clé d' Service AWS identification. L' Service AWS identifiant est basé sur le modèle d'API `serviceId` en remplaçant tous les espaces par des traits de soulignement et en minuscules toutes les lettres. Pour obtenir la liste de toutes les clés d'identification de service à utiliser dans `services` cette section, consultezUtilisation de points de terminaison dans le AWS CLI. La clé d'identification du service est suivie de paramètres imbriqués, chacun sur sa propre ligne et indentés de deux espaces.

L'exemple suivant configure le point de terminaison à utiliser pour les demandes adressées au Amazon DynamoDB service dans la *my-services* section utilisée dans le *dev* profil. Toutes les lignes indentées qui suivent immédiatement sont incluses dans cette sous-section et s'appliquent à ce service.

```
[profile dev]
services = my-services

[services my-services]
dynamodb =
  endpoint_url = http://localhost:8000
```

Pour plus d'informations sur les points de terminaison spécifiques à un service, consultez. Utilisation de points de terminaison dans le AWS CLI

Si votre profil possède des informations d'identification basées sur les rôles configurées via un `source_profile` paramètre pour la fonctionnalité IAM d'assumer un rôle, le SDK utilise uniquement les configurations de service pour le profil spécifié. Il n'utilise pas de profils auxquels des rôles sont liés. Par exemple, en utilisant le `config` fichier partagé suivant :

```
[profile A]
credential_source = Ec2InstanceMetadata
endpoint_url = https://profile-a-endpoint.aws/
```

```
[profile B]
source_profile = A
role_arn = arn:aws:iam::123456789012:role/roleB
services = profileB

[services profileB]
ec2 =
  endpoint_url = https://profile-b-ec2-endpoint.aws
```

Si vous utilisez le profil B et que vous appelez Amazon dans votre code EC2, le point de terminaison est résolu commehttps://profile-b-ec2-endpoint.aws. Si votre code envoie une demande à un autre service, la résolution du point de terminaison ne suivra aucune logique personnalisée. Le point de terminaison ne correspond pas au point de terminaison global défini dans le profilA. Pour qu'un point de terminaison global prenne effet pour B le profil, vous devez le définir endpoint_url directement dans le profilB.

Où les paramètres de configuration sont-ils stockés ?

Le AWS CLI stocke les informations d'identification sensibles que vous spécifiez aws configure dans un fichier local nommécredentials, dans un dossier nommé .aws dans votre répertoire personnel. Les autres options de configuration que vous spécifiez avec aws configure sont stockées dans un fichier local nommé config, également stocké dans le dossier .aws du répertoire de base.

> **Stockage des informations d'identification dans le fichier de configuration**
>
> Vous pouvez conserver tous les paramètres de votre profil dans un seul fichier, car les informations d'identification AWS CLI peuvent être lues dans le config fichier. S'il existe des informations d'identification dans les deux fichiers pour un profil partageant le même nom, les clés du fichier d'informations d'identification ont priorité. Nous vous suggérons de conserver les informations d'identification dans les credentials fichiers. Ces fichiers sont également utilisés par les différents kits de développement de logiciels linguistiques (SDKs). Si vous en utilisez un SDKs en plus du AWS CLI, vérifiez si les informations d'identification doivent être stockées dans leur propre fichier.

L'emplacement du répertoire de base varie selon le système d'exploitation, mais il est référencé à l'aide de variables d'environnement %UserProfile% sous Windows et $HOME ou ~ (tilde) sur les systèmes Unix. Vous pouvez spécifier un emplacement autre que celui par défaut pour les fichiers en

définissant les variables d'`AWS_SHARED_CREDENTIALS_FILE`environnement `AWS_CONFIG_FILE` et sur un autre chemin local. Consultez Configuration des variables d'environnement pour AWS CLI pour plus de détails.

Lorsque vous utilisez un profil partagé qui spécifie un rôle AWS Identity and Access Management (IAM), l'opération AWS CLI appelle l' AWS STS `AssumeRole`opération pour récupérer des informations d'identification temporaires. Ces informations d'identification sont ensuite stockées (dans `~/.aws/cli/cache`). Les AWS CLI commandes suivantes utilisent les informations d'identification temporaires mises en cache jusqu'à leur expiration, puis actualisent AWS CLI automatiquement les informations d'identification.

Utilisation de profils nommés

Si aucun profil n'est défini explicitement, le `default` profil est utilisé.

Pour utiliser un profil nommé, ajoutez l'option `--profile` *`profile-name`* à votre commande. L'exemple suivant répertorie toutes vos EC2 instances Amazon à l'aide des informations d'identification et des paramètres définis dans le `user1` profil.

```
$ aws ec2 describe-instances --profile user1
```

Pour utiliser un profil nommé pour plusieurs commandes, vous pouvez éviter de spécifier le profil dans chaque commande en définissant la variable d'`AWS_PROFILE`environnement comme profil par défaut. Vous pouvez annuler ce paramètre en utilisant le `--profile` paramètre.

Linux or macOS

```
$ export AWS_PROFILE=user1
```

Windows

```
C:\> setx AWS_PROFILE user1
```

L'utilisation de `set` pour définir une variable d'environnement modifie la valeur utilisée jusqu'à la fin de la session d'invite de commande en cours, ou jusqu'à ce que vous définissiez la variable sur une autre valeur.

L'utilisation de `setx` pour définir une variable d'environnement modifie la valeur dans tous les shells de commande que vous créez après l'exécution de la commande. Cela n'affecte aucun

shell de commande qui est déjà en cours d'exécution au moment où vous exécutez la commande. Fermez et redémarrez le shell de commande pour voir les effets de la modification.

La définition de la variable d'environnement permet de modifier le profil par défaut jusqu'à la fin de votre session shell, ou jusqu'à ce que vous définissiez la variable sur une autre valeur. Vous pouvez rendre des variables persistantes dans de futures sessions en les plaçant dans votre script de démarrage de shell. Pour de plus amples informations, veuillez consulter Configuration des variables d'environnement pour AWS CLI.

Définir et afficher les paramètres de configuration à l'aide de commandes

Il existe plusieurs manières d'afficher et de définir vos paramètres de configuration à l'aide de commandes.

`aws configure`

Exécutez cette commande pour définir et afficher rapidement vos informations d'identification, votre région et votre format de sortie. L'exemple suivant montre des exemples de valeurs.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

`aws configure set`

Vous pouvez définir les informations d'identification ou les paramètres de configuration à l'aide de la commande `aws configure set`. Spécifiez le profil que vous souhaitez consulter ou modifier avec le paramètre `--profile`.

Par exemple, la commande suivante définit le paramètre `region` dans le profil nommé `integ`.

```
$ aws configure set region us-west-2 --profile integ
```

Pour supprimer un paramètre, supprimez-le manuellement dans vos `credentials` fichiers `config` et dans un éditeur de texte.

aws configure get

Vous pouvez récupérer les informations d'identification ou les paramètres de configuration que vous avez définis via `aws configure get`. Spécifiez le profil que vous souhaitez consulter ou modifier avec le paramètre `--profile`.

Par exemple, la commande suivante récupère le paramètre `region` dans le profil nommé `integ`.

```
$ aws configure get region --profile integ
us-west-2
```

Un résultat vide indique que le paramètre n'est pas explicitement défini et utilise la valeur par défaut.

aws configure import

Importez CSV les informations d'identification générées depuis la console Web IAM. Cela ne concerne pas les informations d'identification générées par IAM Identity Center ; les clients qui utilisent IAM Identity Center doivent utiliser aws configure sso. Un fichier CSV est importé avec le nom du profil correspondant au nom d'utilisateur. Le fichier CSV doit contenir les en-têtes suivants.

- Nom utilisateur
- ID de clé d'accès
- Clé d'accès secrète

> **Note**
>
> Lors de la création initiale de la paire de clés, une fois que vous avez fermé la boîte de dialogue Télécharger le fichier .csv, vous ne pouvez plus accéder à votre clé d'accès secrète après avoir fermé la boîte de dialogue. Si vous avez besoin d'un `.csv` fichier, vous devez en créer un vous-même avec les en-têtes requis et les informations que vous avez enregistrées sur les paires de clés. Si vous n'avez pas accès aux informations relatives à votre paire de clés, vous devez créer une nouvelle paire de clés.

```
$ aws configure import --csv file://credentials.csv
```

`aws configure list`

Pour répertorier les données de configuration, utilisez la `aws configure list` commande. Cette commande répertorie le profil, la clé d'accès, la clé secrète et les informations de configuration de région utilisées pour le profil spécifié. Pour chaque élément de configuration, il indique la valeur, l'endroit où la valeur de configuration a été récupérée et le nom de la variable de configuration.

Par exemple, si vous fournissez la variable d'environnement Région AWS in an, cette commande indique le nom de la région que vous avez configurée, indique que cette valeur provient d'une variable d'environnement et le nom de la variable d'environnement.

Pour les méthodes d'identification temporaires telles que les rôles et IAM Identity Center, cette commande affiche la clé d'accès temporairement mise en cache et la clé d'accès secrète est affichée.

```
$ aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key     ****************ABCD  shared-credentials-file
secret_key     ****************ABCD  shared-credentials-file
    region                us-west-2             env    AWS_DEFAULT_REGION
```

`aws configure list-profiles`

Pour afficher la liste de tous les noms de profil, utilisez la commande `aws configure list-profiles`.

```
$ aws configure list-profiles
default
test
```

`aws configure sso`

Exécutez cette commande pour définir et afficher rapidement vos AWS IAM Identity Center informations d'identification, votre région et votre format de sortie. L'exemple suivant montre des exemples de valeurs.

```
$ aws configure sso
```

```
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

`aws configure sso-session`

Exécutez cette commande pour définir et afficher rapidement vos AWS IAM Identity Center informations d'identification, votre région et votre format de sortie dans la section sso-session des fichiers `credentials` et`config`. L'exemple suivant montre des exemples de valeurs.

```
$ aws configure sso-session
SSO session name: my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

`aws configure export-credentials`

Exécutez cette commande pour exporter les informations d'identification actuellement définies dans le format spécifié. Par défaut, la commande exporte les informations d'identification par défaut au `process` format, qui est un format JSON pris en charge par le format d'identification AWS SDKs and Tools.

```
$ aws configure export-credentials
{
  "Version": 1,
  "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
}
```

Pour exporter un profil et un format spécifiques, utilisez les `--format` options `--profile` et. Les options de format sont les suivantes :

- (par défaut) **`process`**- Le format JSON pris en charge par la `credential_process` configuration AWS SDKs and Tools.
- **`env`**- Variables d'environnement au format shell exporté.
- **`env-no-export`**- Variables d'environnement non exportées au format shell.
- **`powershell`**- Variables d'environnement au PowerShell format.

- **`windows-cmd`**- Variables d'environnement au format de ligne de commande Windows.

L'exemple suivant exporte le `user1` profil vers un format shell exporté.

```
$ aws configure export-credentials --profile user1 --format env
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Exemples de commandes de définition d'une nouvelle configuration et d'informations d'identification

Les exemples suivants montrent comment configurer un profil par défaut avec des informations d'identification, une région et une sortie spécifiées pour différentes méthodes d'authentification.

IAM Identity Center (SSO)

Cet exemple concerne AWS IAM Identity Center l'utilisation de l'`aws configure sso`assistant. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

Attempting to automatically open the SSO authorization page in your default browser.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

CLI default client Region [None]: us-west-2
CLI default output format [None]: json
```

```
CLI profile name [123456789011_ReadOnly]: user1
```

IAM Identity Center (Legacy SSO)

Cet exemple concerne l'ancienne méthode d' AWS IAM Identity Center utilisation de l'`aws configure sso`assistant. Pour utiliser l'ancienne authentification unique, laissez le nom de session vide. Pour de plus amples informations, veuillez consulter the section called “Authentification IAM Identity Center”.

```
$ aws configure sso
SSO session name (Recommended):
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

CLI default client Region [None]: us-west-2
CLI default output format [None]: json
CLI profile name [123456789011_ReadOnly]: user1
```

Short-term credentials

Cet exemple concerne les informations d'identification à court terme de AWS Identity and Access Management. L'assistant de configuration aws est utilisé pour définir les valeurs initiales, puis la `aws configure set` commande attribue la dernière valeur requise. Pour de plus amples informations, veuillez consulter the section called “Identifiants à court terme”.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
$ aws configure set
 aws_session_token fcZib3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

IAM role

Cet exemple permet d'assumer un rôle IAM. Les profils qui utilisent des rôles IAM extraient les informations d'identification d'un autre profil, puis appliquent les autorisations des rôles IAM. Dans les exemples suivants, `default` est le profil source des informations d'identification et `user1` emprunte les mêmes informations d'identification puis assume un nouveau rôle. Il n'y a pas d'assistant pour ce processus, donc chaque valeur est définie à l'aide de la `aws configure set` commande. Pour de plus amples informations, veuillez consulter the section called “Rôles IAM”.

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
$ aws configure set source_profile default
$ aws configure set role_session_name session_user1
$ aws configure set region us-west-2
$ aws configure set output json
```

Amazon EC2 instance metadata credentials

Cet exemple concerne les informations d'identification obtenues à partir des métadonnées de l' EC2 instance Amazon d'hébergement. Il n'y a pas d'assistant pour ce processus, donc chaque valeur est définie à l'aide de la `aws configure set` commande. Pour de plus amples informations, veuillez consulter the section called “ EC2 Métadonnées Amazon”.

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
$ aws configure set credential_source Ec2InstanceMetadata
$ aws configure set region us-west-2
$ aws configure set output json
```

Long-term credentials

> **Warning**
>
> Afin d’éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous

travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que AWS IAM Identity Center.

Cet exemple concerne les informations d'identification à long terme de AWS Identity and Access Management. Pour de plus amples informations, veuillez consulter the section called “Utilisateurs IAM”.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Paramètres de fichier **`config`** pris en charge

Rubriques

- Global Settings (Paramètres globaux)
- Paramètres de commandes personnalisées S3

Les paramètres suivants sont pris en charge dans le fichier `config`. Les valeurs reprises dans le profil spécifié (ou par défaut) sont utilisées, sauf si elles sont remplacées par la présence d'une variable d'environnement de même nom ou une option de ligne de commande de même nom. Pour plus d'informations sur les paramètres de commande qui sont prédéfinis, veuillez consulter Configuration des paramètres pour AWS CLI

Global Settings (Paramètres globaux)

`account_id_endpoint_mode`

Spécifie s'il faut utiliser un point de terminaison AWS basé sur le compte IDs pour les appels vers le support. Services AWS Pour plus d'informations sur les points de terminaison basés sur des comptes, consultez. the section called “Points de terminaison basés sur des comptes”

Ce paramètre peut être défini comme suit :

- (par défaut) **`preferred`**— Le point de terminaison doit inclure l'ID de compte s'il est disponible.
- **`disabled`**— Un point de terminaison résolu n'inclut pas d'identifiant de compte.

- **`required`**— Le point de terminaison doit inclure l'identifiant du compte. Si l'identifiant du compte n'est pas disponible, le SDK génère une erreur.

Peut être remplacée par la variable d'environnement `AWS_ACCOUNT_ID_ENDPOINT_MODE`. Pour utiliser des points de terminaison basés sur un compte, l'ID doit être défini dans la variable ou le paramètre d'`AWS_ACCOUNT_ID`environnement. `aws_account_id`

```
account_id_endpoint_mode = preferred
```

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called “Configuration du point de terminaison et priorité des paramètres”.

`aws_access_key_id`

Spécifie la clé AWS d'accès utilisée dans le cadre des informations d'identification pour authentifier la demande de commande. Bien qu'elle puisse être stockée dans le fichier `config`, nous vous recommandons de la stocker dans le fichier `credentials`.

Peut être remplacée par la variable d'environnement `AWS_ACCESS_KEY_ID`. Vous ne pouvez pas spécifier l'ID de clé d'accès sous forme d'option de ligne de commande.

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

`aws_account_id`

Spécifie l'ID du point de terminaison AWS basé sur le compte à utiliser pour les appels vers le service pris en charge. Services AWS Pour plus d'informations sur les points de terminaison basés sur des comptes, consultez. the section called “Points de terminaison basés sur des comptes”

Peut être remplacée par la variable d'environnement AWS_ACCOUNT_ID.
La variable ou le account_id_endpoint_mode paramètre d'AWS_ACCOUNT_ID_ENDPOINT_MODEenvironnement doit être défini sur preferred ou required pour utiliser ce paramètre.

```
aws_account_id = 123456789EXAMPLE
```

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

aws_secret_access_key

Spécifie la clé AWS secrète utilisée dans le cadre des informations d'identification pour authentifier la demande de commande. Bien qu'elle puisse être stockée dans le fichier config, nous vous recommandons de la stocker dans le fichier credentials.

Peut être remplacée par la variable d'environnement AWS_SECRET_ACCESS_KEY. Vous ne pouvez pas spécifier la clé d'accès secrète sous forme d'option de ligne de commande.

```
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

aws_session_token

Spécifie un jeton de AWS session. Un jeton de session n'est nécessaire que si vous spécifiez manuellement des informations d'identification de sécurité temporaires. Bien qu'elle puisse être stockée dans le fichier config, nous vous recommandons de la stocker dans le fichier credentials.

Peut être remplacée par la variable d'environnement AWS_SESSION_TOKEN. Vous ne pouvez pas le jeton de session sous forme d'option de ligne de commande.

```
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

ca_bundle

Spécifie un ensemble de certificat CA (un fichier avec l'extension `.pem`) qui est utilisé pour vérifier les certificats SSL.

Peut être remplacé par la variable d'environnement AWS_CA_BUNDLE ou l'option de ligne de commande `--ca-bundle`.

```
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

cli_auto_prompt

Active l'invite automatique pour la AWS CLI version 2. Deux paramètres peuvent être utilisés :

- **on**utilise le mode d'invite automatique complet chaque fois que vous essayez d'exécuter une `aws` commande. Cela inclut d'appuyer sur ENTER après une commande complète ou une commande incomplète.

  ```
  cli_auto_prompt = on
  ```

- **on-partial**utilise le mode d'invite automatique partiel. Si une commande est incomplète ou ne peut pas être exécutée en raison d'erreurs de validation côté client, l'invite automatique est utilisée. Ce mode est particulièrement utile si vous avez des scripts ou des runbooks préexistants, ou si vous souhaitez uniquement être automatiquement invité à saisir des commandes que vous ne connaissez pas au lieu d'être invité à chaque commande.

  ```
  cli_auto_prompt = on-partial
  ```

Vous pouvez remplacer ce paramètre en utilisant la variable d'aws_cli_auto_promptenvironnement ou les paramètres de ligne de `--no-cli-auto-prompt` commande `--cli-auto-prompt` et.

Pour plus d'informations sur la fonctionnalité d'invite automatique de la AWS CLI version 2, consultezActivation et utilisation des invites de commande dans le AWS CLI.

cli_binary_format

Spécifie la manière dont la AWS CLI version 2 interprète les paramètres d'entrée binaires. Il peut avoir l'une des valeurs suivantes :

- base64 — Il s'agit de la valeur par défaut. Un paramètre d'entrée qui est saisi comme un objet BLOB accepte une chaîne codée en base64. Pour transmettre le contenu binaire réel, placez le contenu dans un fichier et indiquez le chemin d'accès et le nom du fichier avec le préfixe `fileb://` comme valeur du paramètre. Pour transmettre du texte codé en base64 contenu dans un fichier, indiquez le chemin d'accès et le nom du fichier avec le préfixe `file://` comme valeur du paramètre.
- raw-in-base64-out — Valeur par défaut pour la AWS CLI version 1. Si la valeur du paramètre est`raw-in-base64-out`, les fichiers référencés à l'aide du `file://` préfixe sont lus sous forme de texte, puis les AWS CLI tentatives de codage en binaire sont tentées.

Cette entrée n'a pas de variable d'environnement équivalente. Vous pouvez spécifier la valeur sur une commande seule à l'aide du paramètre `--cli-binary-format raw-in-base64-out`.

```
cli_binary_format = raw-in-base64-out
```

Si vous référencez une valeur binaire dans un fichier à l'aide de la notation `fileb://` préfixe, le fichier s'attend AWS CLI toujours à ce que le fichier contienne du contenu binaire brut et ne tente pas de convertir la valeur.

Si vous référencez une valeur binaire dans un fichier à l'aide de la notation `file://` préfixe, AWS CLI le fichier est géré conformément au `cli_binary_format` paramètre actuel. Si la valeur de ce paramètre est `base64` (valeur par défaut lorsqu'elle n'est pas définie explicitement), le fichier AWS CLI s'attend à ce que le fichier contienne du texte codé en base64. Si la valeur de ce paramètre est`raw-in-base64-out`, le fichier AWS CLI s'attend à ce que le fichier contienne du contenu binaire brut.

cli_history

Ce paramètre est désactivé par défaut. Ce paramètre active l'historique des commandes pour AWS CLI. Après avoir activé ce paramètre, AWS CLI enregistre l'historique des `aws` commandes.

```
cli_history = enabled
```

Vous pouvez répertorier votre historique à l'aide de la `aws history list` commande et utiliser la `aws history show` commande qui `command_ids` en résulte pour plus de détails. Pour plus d'informations, voir le guide AWS CLI de référence.

`cli_pager`

Spécifie le programme de pager utilisé pour la sortie. Par défaut, AWS CLI la version 2 renvoie toutes les sorties via le programme de téléavertisseur par défaut de votre système d'exploitation.

Peut être remplacée par la variable d'environnement AWS_PAGER.

```
cli_pager=less
```

`cli_timestamp_format`

Spécifie le format de sortie des valeurs d'horodatage. Vous pouvez spécifier chacune des valeurs suivantes :

- iso8601 — La valeur par défaut pour la AWS CLI version 2. Si cela est spécifié, le AWS CLI reformatage de tous les horodatages de la sortie conformément à la norme ISO 8601. Les horodatages au format ISO 8601 ressemblent aux exemples suivants. L'exemple suivant montre comment l'heure est formatée en séparant la date et l'heure par un T et en incluant un Z après l'heure.

  ```
  YYYY-MM-DDThh:mm:ssZ
  ```

  Les exemples suivants montrent un horodatage utilisant le formatage précédent.

  ```
  2024-05-08T15:16:43Z
  ```

- wire — La valeur par défaut pour la AWS CLI version 1. Si cela est spécifié, AWS CLI affiche toutes les valeurs d'horodatage exactement telles qu'elles ont été reçues dans la réponse à la requête HTTP.

Ce paramètre ne possède pas de variable d'environnement ou d'option de ligne de commande équivalente. Ce paramètre ne modifie pas les entrées d'horodatage, uniquement le formatage des sorties.

```
cli_timestamp_format = iso8601
```

`credential_process`

Spécifie une commande externe à exécuter AWS CLI pour générer ou récupérer les informations d'authentification à utiliser pour cette commande. La commande doit renvoyer les informations d'identification dans un format spécifique. Pour plus d'informations sur l'utilisation de ce paramètre, consultez Obtenir des informations d'identification par le biais d'un processus externe dans le AWS CLI.

Cette entrée n'a pas d'équivalent variable d'environnement ou option de ligne de commande.

```
credential_process = /opt/bin/awscreds-retriever --username susan
```

`credential_source`

Utilisé dans les EC2 instances ou les conteneurs Amazon pour spécifier où se AWS CLI trouvent les informations d'identification à utiliser pour assumer le rôle que vous avez spécifié avec le `role_arn` paramètre. Vous ne pouvez pas spécifier à la fois `source_profile` et `credential_source` dans le même profil.

Ce paramètre peut avoir l'une des trois valeurs suivantes :

- Environnement — Spécifie qu'il s' AWS CLI agit de récupérer les informations d'identification de la source à partir des variables d'environnement.
- Ec2 InstanceMetadata — Spécifie qu' AWS CLI il faut utiliser le rôle IAM attaché au profil d'EC2 instance pour obtenir les informations d'identification de la source.
- EcsContainer— Spécifie qu' AWS CLI il faut utiliser le rôle IAM attaché au conteneur ECS comme informations d'identification source.

```
credential_source = Ec2InstanceMetadata
```

`duration_seconds`

Spécifie la durée maximale de la session de rôle, en secondes. La valeur peut être comprise entre 900 secondes (15 minutes) et la valeur de durée de session maximale définie pour le rôle (soit 43 200 au maximum). Il s'agit d'un paramètre facultatif et, par défaut, la valeur est définie sur 3600 secondes.

`endpoint_url`

Spécifie le point de terminaison utilisé pour toutes les demandes de service. Si ce paramètre est utilisé dans la `services`section du `config` fichier, le point de terminaison est utilisé uniquement

pour le service spécifié. Pour de plus amples informations, veuillez consulter the section called “Définir un point de terminaison global pour tous Services AWS”.

L'exemple suivant utilise le point de terminaison global `http://localhost:1234` et un point de terminaison spécifique au service `http://localhost:4567` pour Amazon S3.

```
[profile dev]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
  endpoint_url = http://localhost:4567
```

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called “Configuration du point de terminaison et priorité des paramètres”.

`ignore_configure_endpoint_urls`

Si cette option est activée, elle AWS CLI ignore toutes les configurations de point de terminaison personnalisées spécifiées dans le `config` fichier. Les valeurs valides sont **`true`** et **`false`**.

```
ignore_configure_endpoint_urls = true
```

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus

élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

`external_id`

Identifiant unique utilisé par des tiers pour assumer un rôle dans les comptes de leurs clients. Il est mappé au paramètre `ExternalId` dans l'opération `AssumeRole`. Ce paramètre n'est nécessaire que si la stratégie d'approbation du rôle spécifie une valeur pour `ExternalId`. Pour plus d'informations, consultez la section Comment utiliser un identifiant externe pour accorder l'accès à vos AWS ressources à un tiers dans le guide de l'utilisateur IAM.

`max_attempts`

Spécifie la valeur maximale de tentatives utilisées par le gestionnaire de AWS CLI nouvelles tentatives, l'appel initial étant pris en compte dans la `max_attempts` valeur que vous fournissez.

Vous pouvez remplacer cette valeur en utilisant la variable d'`AWS_MAX_ATTEMPTS`environnement.

```
max_attempts = 3
```

`mfa_serial`

Numéro d'identification d'un périphérique MFA à utiliser lors de l'endossement d'un rôle. Cela est obligatoire seulement si la stratégie d'approbation du rôle assumé inclut une condition exigeant une authentification MFA. La valeur peut être un numéro de série pour un périphérique matériel (tel que `GAHT12345678`) ou un Amazon Resource Name (ARN) pour un périphérique MFA virtuel (tel que `arn:aws:iam::123456789012:mfa/`*`user`*).

`output`

Spécifie le format de sortie par défaut des commandes demandées en utilisant ce profil. Vous pouvez spécifier les valeurs suivantes :

- **`json`**— La sortie est formatée sous forme de chaîne JSON.
- **`yaml`**— La sortie est formatée sous forme de chaîne YAML.
- **`yaml-stream`**— La sortie est diffusée et formatée sous forme de chaîne YAML. Le streaming permet de traiter plus rapidement de gros types de données.
- **`text`**— La sortie est formatée sous la forme de plusieurs lignes de valeurs de chaîne séparées par des tabulations. Cela peut être utile pour transmettre le résultat à un processeur de texte, comme `grep`sed, ou`awk`.

- **`table`**— La sortie est formatée sous forme de tableau en utilisant les caractères +|- pour former les bordures des cellules. La présentation des informations est dans un format beaucoup plus lisible par l'utilisateur que les autres, mais peu pratique du point de vue programmation.

Peut être remplacé par la variable d'environnement `AWS_DEFAULT_OUTPUT` ou l'option de ligne de commande `--output`.

```
output = table
```

`parameter_validation`

Spécifie si le AWS CLI client tente de valider les paramètres avant de les envoyer au point de terminaison du AWS service.

- true — Il s'agit de la valeur par défaut. Si cela est spécifié, il AWS CLI effectue une validation locale des paramètres de ligne de commande.
- false — Si spécifié, les paramètres de ligne de commande AWS CLI ne sont pas validés avant de les envoyer au point de terminaison du AWS service.

Cette entrée n'a pas d'équivalent variable d'environnement ou option de ligne de commande.

```
parameter_validation = false
```

`region`

Spécifie Région AWS à qui envoyer les demandes pour les commandes demandées à l'aide de ce profil.

- Vous pouvez spécifier n'importe quel code de région disponible pour le service choisi, comme indiqué dans la section AWS Régions et points de terminaison du Référence générale d'Amazon Web Services.
- `aws_globalvous` permet de spécifier le point de terminaison global pour les services qui prennent en charge un point de terminaison mondial en plus des points de terminaison régionaux, tels que AWS Security Token Service (AWS STS) et Amazon Simple Storage Service (Amazon S3).

Vous pouvez remplacer cette valeur en utilisant la variable d'`AWS_REGIO`Nenvironnement, la variable d'`AWS_DEFAULT_REGIO`Nenvironnement ou l'option de ligne de `--region` commande.

```
region = us-west-2
```

`request_checksum_calculation`

Spécifie le moment où une somme de contrôle est calculée pour les charges utiles des demandes et propose les options suivantes :

- `when_supported`— (Par défaut) La somme de contrôle de la charge utile de la demande est calculée lorsqu'une opération spécifie un algorithme de somme de contrôle dans son modèle de service ou nécessite des sommes de contrôle des demandes.
- `when_required`— La somme de contrôle de la charge utile de la demande est calculée lorsqu'une opération nécessite des sommes de contrôle de demande ou lorsqu'un utilisateur fournit une somme de contrôle `requestAlgorithmMember` modélisée par le. Service AWS

```
request_checksum_calculation = when_supported
```

La variable d'environnement `AWS_REQUEST_CHECKSUM_CALCULATION`remplace ce paramètre.

`response_checksum_validation`

Spécifie à quel moment la validation de la somme de contrôle est effectuée pour les charges utiles de réponse et propose les options suivantes :

- `when_supported`— (Par défaut) La validation de la somme de contrôle de la charge utile de réponse est effectuée lorsqu'une opération spécifie un algorithme de réponse dans son modèle de service qu'elle prend en charge. AWS CLI
- `when_required`— La validation de la somme de contrôle de la charge utile de réponse est effectuée lorsqu'une opération spécifie un algorithme de réponse dans son modèle de service qu'elle prend en AWS CLI charge, et que vous définissez le modèle sur lequel il est modélisé `ENABLED` dans l'`requestValidationModeMember`entrée de l'opération.

```
response_checksum_validation = when_supported
```

La variable d'environnement `AWS_RESPONSE_CHECKSUM_VALIDATION`remplace ce paramètre.

`retry_mode`

Spécifie le mode de nouvelle tentative AWS CLI utilisé. Trois modes de nouvelle tentative sont disponibles : `standard` (par défaut), `legacy` (par défaut) et`adaptive`. Pour plus d'informations sur les nouvelles tentatives, consultez la section AWS CLI réessaie dans le AWS CLI.

Vous pouvez remplacer cette valeur en utilisant la variable d'`AWS_RETRY_MODE`environnement.

```
retry_mode = standard
```

role_arn

Spécifie le nom de ressource Amazon (ARN) d'un rôle IAM que vous souhaitez utiliser pour exécuter les AWS CLI commandes. Vous devez également spécifier l'un des paramètres suivants pour identifier les informations d'identification autorisées à assumer ce rôle :

- source_profile
- credential_source

```
role_arn = arn:aws:iam::123456789012:role/role-name
```

La variable d'environnement AWS_ROLE_ARNremplace ce paramètre.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called “Utilisation d'un rôle à l'aide d'une identité web”.

role_session_name

Spécifie le nom à attacher à la session de rôle. Cette valeur est fournie au paramètre `RoleSessionName` lorsque l' AWS CLI appelle l'opération `AssumeRole` et fait partie de l'ARN utilisateur du rôle assumé : `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Ce paramètre est facultatif. Si vous ne spécifiez pas cette valeur, un nom de session est généré automatiquement. Ce nom s'affiche dans les journaux AWS CloudTrail pour les entrées associées à cette session.

```
role_session_name = maria_garcia_role
```

La variable d'environnement AWS_ROLE_SESSION_NAMEremplace ce paramètre.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called “Utilisation d'un rôle à l'aide d'une identité web”.

services

Spécifie la configuration du service à utiliser pour votre profil.

```
[profile dev-s3-specific-and-global]
```

```
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
  endpoint_url = http://localhost:4567
```

Pour plus d'informations sur `services` cette section, consultezthe section called “`services`”.

La variable d'environnement `AWS_ROLE_SESSION_NAME`remplace ce paramètre.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called “Utilisation d'un rôle à l'aide d'une identité web”.

`sdk_ua_app_id`

Un seul Compte AWS peut être utilisé par plusieurs applications clients pour passer des appels à Services AWS. L'ID d'application identifie l'application source qui a effectué un ensemble d'appels à l'aide d'un Service AWS. AWS SDKs et les services n'utilisent ni n'interprètent cette valeur autrement que pour la réintégrer dans les communications avec les clients. Par exemple, cette valeur peut être incluse dans les e-mails opérationnels pour identifier de manière unique laquelle de vos applications est associée à la notification.

L'ID de l'application est une chaîne d'une longueur maximale de 50 caractères. Les lettres, les chiffres et les caractères spéciaux suivants sont autorisés : `! $ % & * + - . , ^ _ ` | ~` Par défaut, aucune valeur n'est attribuée.

```
sdk_ua_app_id = prod1
```

Ce paramètre peut être remplacé à l'aide de la variable d'environnement `AWS_SDK_UA_APP_ID`. Vous ne pouvez pas définir cette valeur en tant que paramètre de ligne de commande.

`sigv4a_signing_region_set`

Spécifie les régions à utiliser lors de la signature avec SigV4a à l'aide d'une liste séparée par des virgules. Si cette variable n'est pas définie, elle AWS CLI utilise la valeur par défaut utilisée par le Service AWS. S'il n'y Service AWS a pas de valeur par défaut, la signature de la demande devient valide dans toutes les régions en utilisant une valeur de`*`.

```
sigv4a_signing_region_set = us-west-2, us-east-1
```

Pour plus d'informations sur SIGv4a, voir AWS Signature version 4 pour les demandes d'API dans le guide de l'utilisateur IAM

Ce paramètre peut être remplacé à l'aide de la variable d'environnement `AWS_SIGV4A_SIGNING_REGION_SET`. Vous ne pouvez pas définir cette valeur en tant que paramètre de ligne de commande.

`source_profile`

Spécifie un profil nommé avec des informations d'identification à long terme que l' AWS CLI peut utiliser pour assumer un rôle que vous avez spécifié avec le paramètre `role_arn`. Vous ne pouvez pas spécifier à la fois `source_profile` et `credential_source` dans le même profil.

```
source_profile = production-profile
```

`sso_account_id`

Spécifie l'ID de AWS compte qui contient le rôle IAM avec l'autorisation que vous souhaitez accorder à l'utilisateur IAM Identity Center associé.

Ce paramètre n'a pas de variable d'environnement ou d'option de ligne de commande.

```
sso_account_id = 123456789012
```

`sso_region`

Spécifie la AWS région qui contient l'hôte du portail d' AWS accès. Cette région est distincte, et peut être différente du paramètre `region` de l'interface de ligne de commande par défaut.

Ce paramètre n'a pas de variable d'environnement ou d'option de ligne de commande.

```
sso_region = us_west-2
```

`sso_registration_scopes`

Une liste séparée par des virgules de portées à autoriser pour le. `sso-session` Les étendues autorisent l'accès aux points de terminaison autorisés par le porteur du jeton IAM Identity Center. Une étendue valide est une chaîne, telle que`sso:account:access`. Ce paramètre ne s'applique pas à l'ancienne configuration non actualisable.

```
sso_registration_scopes = sso:account:access
```

sso_role_name

Spécifie le nom convivial du rôle IAM qui définit les autorisations de l'utilisateur lors de l'utilisation de ce profil.

Ce paramètre n'a pas de variable d'environnement ou d'option de ligne de commande.

```
sso_role_name = ReadAccess
```

sso_start_url

Spécifie l'URL qui pointe vers le portail d' AWS accès de l'organisation. Il AWS CLI utilise cette URL pour établir une session avec le service IAM Identity Center afin d'authentifier ses utilisateurs. Pour trouver l'URL de votre portail d' AWS accès, utilisez l'une des méthodes suivantes :

- Ouvrez votre e-mail d'invitation, l'URL du portail AWS d'accès est répertoriée.
- Ouvrez la AWS IAM Identity Center console à l'adresse https://console.aws.amazon.com/singlesignon/. L'URL du portail d' AWS accès est répertoriée dans vos paramètres.

Ce paramètre n'a pas de variable d'environnement ou d'option de ligne de commande.

```
sso_start_url = https://my-sso-portal.awsapps.com/start
```

use_dualstack_endpoint

Permet d'utiliser des points de terminaison à double pile pour envoyer AWS des demandes. Pour en savoir plus sur les points de terminaison à double pile, qui prennent en charge à la fois le IPv6 trafic IPv4 et le trafic, consultez la section Utilisation des points de terminaison à double pile Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service. Des points de terminaison à double pile sont disponibles pour certains services dans certaines régions. S'il n'existe pas de point de terminaison à double pile pour le service ou Région AWS si la demande échoue. Les paramètres valides sont `true` et`false`. Par défaut, l'option est désactivée. Pour de plus amples informations, veuillez consulter the section called “Conçu pour utiliser des points de terminaison à double pile pour tous Services AWS”.

Ce paramètre et `use_accelerate_endpoint` sont mutuellement exclusifs.

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

`use_fips_endpoint`

Certains AWS services proposent des terminaux compatibles avec la norme fédérale de traitement de l'information (FIPS) 140-2 dans certains cas. Régions AWS Lorsque le AWS service prend en charge le protocole FIPS, ce paramètre indique le point de terminaison FIPS à utiliser AWS CLI . Contrairement aux points de AWS terminaison standard, les points de terminaison FIPS utilisent une bibliothèque logicielle TLS conforme à la norme FIPS 140-2. Ces points de terminaison peuvent être requis par les entreprises qui interagissent avec le gouvernement des États-Unis. Pour plus d'informations, voir,the section called "Paramétré pour utiliser les FIPs points de terminaison pour tous Services AWS".

Si ce paramètre est activé, mais qu'il n'existe aucun point de terminaison FIPS pour le service dans votre ordinateur Région AWS, la AWS commande risque d'échouer. Dans ce cas, spécifiez manuellement le point de terminaison à utiliser dans la commande à l'aide de l'`--endpoint-url`option ou utilisez des points de terminaison spécifiques au service.

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

`web_identity_token_file`

Spécifie le chemin d'accès à un fichier contenant un jeton d'accès OAuth 2.0 ou un jeton d'identification OpenID Connect fourni par un fournisseur d'identité. L' AWS CLI charge le contenu de ce fichier et le transmet en tant qu'argument `WebIdentityToken` à l'opération `AssumeRoleWithWebIdentity`.

La variable d'environnement `AWS_WEB_IDENTITY_TOKEN_FILE` remplace ce paramètre.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called "Utilisation d'un rôle à l'aide d'une identité web".

`tcp_keepalive`

Spécifie si le AWS CLI client utilise des paquets TCP Keep-Alive.

Cette entrée n'a pas d'équivalent variable d'environnement ou option de ligne de commande.

```
tcp_keepalive = false
```

Paramètres de commandes personnalisées S3

Amazon S3 prend en charge plusieurs paramètres qui configurent la manière dont AWS CLI les opérations Amazon S3 sont effectuées. Certains s'appliquent à toutes les commandes S3 dans les deux espaces de noms `s3` et `s3api`. D'autres sont spécifiquement destinées aux commandes « personnalisées » S3 qui font abstraction des opérations courantes et font plus qu'un one-to-one mappage vers une opération d'API. Les commandes de transfert `aws s3 cp`, `sync`, `mv` et `rm` ont des paramètres supplémentaires que vous pouvez utiliser pour contrôler les transferts S3.

Toutes ces options peuvent être configurées en spécifiant le paramètre imbriqué `s3` dans votre fichier `config`. Chaque paramètre est en retrait sur sa propre ligne.

> **Note**
>
> Ces paramètres sont entièrement facultatifs. Vous devriez pouvoir utiliser les commandes de transfert `aws s3` sans configurer aucun de ces paramètres. Ces paramètres sont fournis pour vous permettre d'optimiser les performances ou pour prendre en compte l'environnement spécifique dans lequel vous exécutez ces commandes `aws s3`.

Ces paramètres sont tous définis sous une clé `s3` de niveau supérieur dans le fichier `config`, comme illustré dans l'exemple suivant pour le profil `development` :

```
[profile development]
s3 =
  max_concurrent_requests = 20
  max_queue_size = 10000
  multipart_threshold = 64MB
  multipart_chunksize = 16MB
  max_bandwidth = 50MB/s
  use_accelerate_endpoint = true
  addressing_style = path
```

Les paramètres suivants s'appliquent à n'importe quelle commande S3 dans les espaces de noms `s3` et `s3api`.

`addressing_style`

Spécifie le type d'adressage à utiliser. Il vérifie si le nom du compartiment figure dans le nom d'hôte ou fait partie de l'URL. Les valeurs valides sont `path`, `virtual` et `auto`. La valeur par défaut est `auto`.

Il existe deux styles de construction d'un point de terminaison Amazon S3. Le premier est appelé `virtual` et inclut le nom du compartiment dans le nom d'hôte. Par exemple : `https://`*`bucketname`*`.s3.amazonaws.com`. Sinon, avec le style `path`, vous traitez le nom du compartiment comme s'il s'agissait d'un chemin d'accès dans l'URI : par exemple, `https://s3.amazonaws.com/`*`bucketname`*. La valeur par défaut dans l'interface de ligne de commande consiste à utiliser `auto`, qui tente d'utiliser le style `virtual` chaque fois que possible, mais bascule, au besoin, sur le style `path`. Par exemple, si le nom de votre compartiment n'est pas compatible DNS, ce nom ne peut pas faire partie du nom d'hôte et doit être dans le chemin d'accès. Avec `auto`, l'interface de ligne de commande détecte cette condition et passe automatiquement au style `path`. Si vous définissez le style d'adressage sur`path`, vous devez alors vous assurer que la AWS région dans laquelle vous avez configuré AWS CLI correspond à la région de votre compartiment.

`payload_signing_enabled`

Spécifie s'il faut SHA256 signer les charges utiles sigv4. Par défaut, cette option est désactivée pour les chargements de streaming (`UploadPart` et `PutObject`) lors de l'utilisation de https. Par défaut, cette option est définie sur `false` pour les chargements de streaming (`UploadPart` et

`PutObject`), mais uniquement si un `ContentMD5` est présent (il est générée par défaut) et sir le point de terminaison utilise le protocole HTTPS.

Si ce paramètre est défini sur true, les demandes S3 reçoivent une validation de contenu supplémentaire sous la forme d'une SHA256 somme de contrôle calculée pour vous et incluse dans la signature de la demande. Si la valeur est false, le total de contrôle n'est pas calculé. La désactivation peut s'avérer utile pour réduire la surcharge en termes de performances générée par le calcul du total de contrôle.

`use_accelerate_endpoint`

Utilisez le point de terminaison Amazon S3 Accelerate pour toutes les `s3api` commandes `s3` et. La valeur par défaut est false. Ce paramètre et `use_dualstack_endpoint` sont mutuellement exclusifs.

S'il est défini sur true, toutes les AWS CLI demandes Amazon S3 sont dirigées vers le `S3 Accelerate` point de terminaison situé à`s3-accelerate.amazonaws.com`. Pour utiliser ce point de terminaison, vous devez activer votre compartiment pour utiliser `S3 Accelerate`. Toutes les requêtes sont envoyées en utilisant le style virtuel de l'adressage de compartiment : *`my-bucket`*`.s3-accelerate.amazonaws.com`. Les requêtes `ListBuckets`, `CreateBucket` et `DeleteBucket` ne sont pas envoyées au point de terminaison S3 Accelerate, car celui-ci ne prend pas en charge ces opérations. Ce comportement peut également être défini sur si le paramètre `--endpoint-url` est défini sur `https://s3-accelerate.amazonaws.com` ou `http://s3-accelerate.amazonaws.com` pour toute commande `s3api` ou `s3`.

`use_dualstack_endpoint`

Permet d'utiliser des points de terminaison à double pile pour les envois `s3` et `s3api` les demandes. Pour en savoir plus sur les points de terminaison à double pile, qui prennent en charge à la fois le IPv6 trafic IPv4 et le trafic, consultez la section Utilisation des points de terminaison à double pile Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service. Des points de terminaison à double pile sont disponibles pour certains services dans certaines régions. S'il n'existe pas de point de terminaison à double pile pour le service ou Région AWS si la demande échoue. Les paramètres valides sont `true` et`false`. Par défaut, l'option est désactivée. Pour de plus amples informations, veuillez consulter the section called “Conçu pour utiliser des points de terminaison à double pile pour tous Services AWS”.

Ce paramètre et `use_accelerate_endpoint` sont mutuellement exclusifs.

Les paramètres suivants ne s'appliquent qu'aux commandes de l'ensemble de commandes de l'espace de noms `s3` :

`max_bandwidth`

Spécifie la bande passante maximale qui peut être consommée pour le chargement et le téléchargement de données vers et depuis Amazon S3. La valeur par défaut sans limite.

Cela limite la bande passante maximale que les commandes S3 peuvent utiliser pour transférer des données vers et depuis Amazon S3. Cette valeur ne s'applique qu'aux chargements et aux téléchargements ; elle ne s'applique pas aux copies ou suppressions. La valeur est exprimée en octets par seconde. La valeur peut être spécifiée comme suit :

- Un entier. Par exemple, `1048576` définit l'utilisation maximale de bande passante à 1 méga-octet par seconde.
- Un entier suivi d'un taux suffixe de débit. Vous pouvez spécifier des suffixes de débit en utilisant : `KB/s`, `MB/s`, ou `GB/s`. Par exemple, `300KB/s`, `10MB/s`.

En général, nous vous recommandons de commencer par essayer de réduire la consommation de bande passante en diminuant `max_concurrent_requests`. Si cela ne limite pas la consommation de bande passante de façon adéquate pour le débit souhaité, vous pouvez utiliser le paramètre `max_bandwidth` pour limiter davantage la consommation de bande passante. Il en est ainsi parce que `max_concurrent_requests` contrôle le nombre de threads en cours d'exécution. Si à la place vous baissez d'abord `max_bandwidth`, mais conservez une valeur `max_concurrent_requests` élevée, cela peut conduire certains threads à attendre inutilement. Cela peut entraîner une consommation excessive de ressources et des délais d'expiration de connexion.

`max_concurrent_requests`

Spécifie le nombre maximal de requêtes simultanées. La valeur par défaut est 10.

Le commandes de transfert `aws s3` sont multithread. À tout moment, plusieurs requêtes Amazon S3 peuvent être en cours d'exécution. Par exemple, lorsque vous utilisez la commande `aws s3 cp localdir s3://bucket/ --recursive` pour télécharger des fichiers dans un compartiment S3, ils AWS CLI peuvent télécharger les fichiers `localdir/file1localdir/file2`, et `localdir/file3` en parallèle. Le paramètre `max_concurrent_requests` spécifie le nombre maximal d'opérations de transfert qui peuvent s'exécuter en même temps.

Vous devrez peut-être modifier cette valeur pour quelques raisons :

- Diminution de cette valeur — Dans certains environnements, la valeur par défaut de 10 demandes simultanées peut submerger le système. Cela peut entraîner des délais de connexion ou réduire la réactivité du système. La réduction de cette valeur rend les commandes de transfert S3 moins gourmandes en ressources. Le compromis est que les transferts S3 peuvent prendre plus de temps à s'exécuter. La réduction de cette valeur peut s'avérer nécessaire si vous utilisez un outil pour limiter la bande passante.
- Augmenter cette valeur — Dans certains scénarios, vous souhaiterez peut-être que les transferts Amazon S3 soient effectués le plus rapidement possible, en utilisant autant de bande passante réseau que nécessaire. Dans ce scénario, le nombre par défaut de requêtes simultanées peut ne pas être suffisant pour utiliser toute la bande passante réseau disponible. L'augmentation de cette valeur peut améliorer le temps nécessaire pour effectuer un transfert Amazon S3.

`max_queue_size`

Spécifie le nombre maximal de tâches dans la file d'attente de tâches. La valeur par défaut est 1000.

AWS CLI En interne, il utilise un modèle dans lequel il met en file d'attente les tâches Amazon S3 qui sont ensuite exécutées par des consommateurs dont le nombre est limité à. `max_concurrent_requests` Une tâche correspond généralement à une seule opération Amazon S3. Par exemple, une tâche pourrait être une `PutObjectTask`, `GetObjectTask` ou `UploadPartTask`. La vitesse à laquelle les tâches sont ajoutées à la file d'attente peut être largement plus rapide que celle à laquelle les consommateurs terminent les tâches. Pour éviter une croissance sans limite, la taille de la file d'attente de tâches est limitée à une valeur spécifique. Ce paramètre modifie la valeur de ce nombre maximal.

En général, vous n'avez pas besoin de modifier ce paramètre. Ce paramètre correspond également au nombre de tâches dont il a connaissance et qui doivent être exécutées. AWS CLI Cela signifie que par défaut, ils ne AWS CLI peuvent voir que 1000 tâches à venir. L' AWS CLI augmentation de cette valeur permet de connaître plus rapidement le nombre total de tâches nécessaires, en supposant que le taux de mise en file d'attente est plus rapide que le taux d'achèvement des tâches. Le compromis est qu'une max_queue_size plus importante nécessite plus de mémoire.

`multipart_chunksize`

Spécifie la taille de bloc AWS CLI utilisée pour les transferts partitionnés de fichiers individuels. La valeur par défaut est 8 Mo, avec un minimum de 5 Mo.

Lorsqu'un transfert de fichier dépasse le`multipart_threshold`, il AWS CLI divise le fichier en morceaux de cette taille. Cette valeur peut être spécifiée en utilisant la même syntaxe que `multipart_threshold`, comme nombre d'octets sous la forme d'un entier ou en utilisant une taille et un suffixe.

`multipart_threshold`

Spécifie le seuil de taille AWS CLI utilisé pour les transferts partitionnés de fichiers individuels. La valeur par défaut est de 8 Mo.

Lors du chargement, du téléchargement ou de la copie d'un fichier, les commandes Amazon S3 passent aux opérations en plusieurs parties si le fichier dépasse cette taille. Vous pouvez spécifier cette valeur de deux manières :

- La taille de fichier en octets. Par exemple, `1048576`.
- La taille de fichier avec un suffixe de taille. Vous pouvez utiliser `KB`, `MB`, `GB` ou `TB`. Par exemple, `10MB`, `1GB`.

> **Note**
>
> S3 peut imposer des contraintes sur les valeurs valides utilisables être utilisées pour les opérations partitionnées. Pour plus d'informations, consultez la documentation relative au chargement partitionné S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Configuration des variables d'environnement pour AWS CLI

Les variables d'environnement constituent un autre moyen de spécifier les options de configuration et les informations d'identification, et peuvent être utiles pour la création de scripts.

Priorité d'options

- Si vous spécifiez une option à l'aide d'une des variables d'environnement décrites dans cette rubrique, elle remplace toute valeur chargée à partir d'un profil dans le fichier de configuration.
- Si vous spécifiez une option à l'aide d'un paramètre sur la ligne de AWS CLI commande, elle remplace toute valeur de la variable d'environnement correspondante ou d'un profil du fichier de configuration.

Pour plus d'informations sur la priorité et la manière dont elle AWS CLI détermine les informations d'identification à utiliser, consultezConfiguration des paramètres pour AWS CLI.

Rubriques

- Comment définir les variables d'environnement
- AWS CLI variables d'environnement prises en charge

Comment définir les variables d'environnement

Les exemples suivants montrent comment vous pouvez configurer des variables d'environnement pour l'utilisateur par défaut.

Linux or macOS

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_REGION=us-west-2
```

La définition de la variable d'environnement permet de modifier la valeur utilisée jusqu'à la fin de votre session shell, ou jusqu'à ce que vous définissiez la variable sur une autre valeur. Vous pouvez rendre les variables persistantes dans de futures sessions en les définissant dans votre script de démarrage de shell.

Windows Command Prompt

À configurer pour toutes les sessions

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx AWS_REGION us-west-2
```

L'utilisation de `setx` pour définir une variable d'environnement modifie la valeur utilisée dans la session d'invite de commande en cours et toutes les sessions d'invite de commande que vous créez après l'exécution de la commande. Cela n'affecte pas les autres shells de commande qui sont déjà en cours d'exécution lorsque vous exécutez la commande. Il se peut que vous deviez redémarrer votre terminal pour que les paramètres se chargent.

À configurer pour la session en cours uniquement

L'utilisation de `set` pour définir une variable d'environnement modifie la valeur utilisée jusqu'à la fin de la session d'invite de commande en cours, ou jusqu'à ce que vous définissiez la variable sur une autre valeur.

```
C:\> set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
C:\> set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> set AWS_REGION=us-west-2
```

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:\> $Env:AWS_REGION="us-west-2"
```

Si vous définissez une variable d'environnement à l' PowerShell invite, comme indiqué dans les exemples précédents, elle enregistre la valeur uniquement pendant la durée de la session en cours. Pour que le paramètre de variable d'environnement soit persistant dans toutes les sessions PowerShell et dans les sessions d'invite de commande, stockez-le à l'aide de l'application système du Panneau de configuration. Vous pouvez également définir la variable pour toutes les PowerShell sessions futures en l'ajoutant à votre PowerShell profil. Consultez la PowerShell documentation pour plus d'informations sur le stockage des variables d'environnement ou leur persistance d'une session à l'autre.

Si des informations d'identification sont configurées dans vos `credentials` fichiers `config` OR, vous pouvez les exporter vers un format de variable d'environnement à l'aide de la `aws configure export-credentials`commande.

AWS CLI variables d'environnement prises en charge

AWS CLI prend en charge les variables d'environnement suivantes.

`AWS_ACCESS_KEY_ID`

Spécifie une clé AWS d'accès associée à un compte IAM.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `aws_access_key_id`. Vous ne pouvez pas spécifier l'ID de clé d'accès en utilisant une option de ligne de commande.

`AWS_ACCOUNT_ID`

Spécifie l'ID du point de terminaison AWS basé sur le compte à utiliser pour les appels vers le service pris en charge. Services AWS Pour plus d'informations sur les points de terminaison basés sur des comptes, consultez. the section called “Points de terminaison basés sur des comptes”

Ce paramètre remplace le `aws_account_id` paramètre. La variable ou le `account_id_endpoint_mode` paramètre d'`AWS_ACCOUNT_ID_ENDPOINT_MODE`environnement doit être défini sur `preferred` ou `required` pour utiliser ce paramètre.

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called “Configuration du point de terminaison et priorité des paramètres”.

`AWS_ACCOUNT_ID_ENDPOINT_MODE`

Spécifie s'il faut utiliser un point de terminaison AWS basé sur le compte IDs pour les appels vers le support. Services AWS Pour plus d'informations sur les points de terminaison basés sur des comptes, consultez. the section called “Points de terminaison basés sur des comptes”

Ce paramètre peut être défini comme suit :

- (par défaut) **`preferred`**— Le point de terminaison doit inclure l'ID de compte s'il est disponible.
- **`disabled`**— Un point de terminaison résolu n'inclut pas d'identifiant de compte.
- **`required`**— Le point de terminaison doit inclure l'identifiant du compte. Si l'identifiant du compte n'est pas disponible, le SDK génère une erreur.

Ce paramètre remplace le `account_id_endpoint_mode` paramètre. Pour utiliser des points de terminaison basés sur un compte, l'ID doit être défini dans la variable ou le paramètre d'`AWS_ACCOUNT_ID`environnement. `aws_account_id`

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

`AWS_CA_BUNDLE`

Spécifie le chemin d'accès vers un ensemble de certificats à utiliser pour la validation des certificats HTTPS.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `ca_bundle`. Vous pouvez remplacer cette variable d'environnement à l'aide du paramètre de ligne de commande `--ca-bundle`.

`AWS_CLI_AUTO_PROMPT`

Active l'invite automatique pour la AWS CLI version 2. Deux paramètres peuvent être utilisés :

- **`on`**utilise le mode d'invite automatique complet chaque fois que vous essayez d'exécuter une `aws` commande. Cela inclut d'appuyer sur ENTER après une commande complète ou une commande incomplète.
- **`on-partial`**utilise le mode d'invite automatique partiel. Si une commande est incomplète ou ne peut pas être exécutée en raison d'erreurs de validation côté client, l'invite automatique est utilisée. Ce mode est utile si vous avez des scripts ou des runbooks préexistants, ou si vous souhaitez uniquement être automatiquement invité à saisir des commandes que vous ne connaissez pas au lieu d'être invité à chaque commande.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de `cli_auto_prompt` profil. Vous pouvez remplacer cette variable d'environnement à l'aide des paramètres de ligne de `--no-cli-auto-prompt` commande `--cli-auto-prompt` et. Pour une liste des encodages standard, consultez la section Encodages standard dans la documentation Python.

Pour plus d'informations sur la fonctionnalité d'invite automatique de la AWS CLI version 2, consultezActivation et utilisation des invites de commande dans le AWS CLI.

AWS_CLI_FILE_ENCODING

Spécifie l'encodage utilisé pour les fichiers texte. Par défaut, l'encodage correspond à vos paramètres régionaux. Pour définir un encodage différent des paramètres régionaux, utilisez la variable d'environnement `AWS_CLI_FILE_ENCODING`. Par exemple, si vous utilisez Windows avec l'encodage `CP1252` par défaut, le paramètre `aws_cli_file_encoding=UTF-8` définit l'interface de ligne de commande de sorte que celle-ci ouvre les fichiers texte au format `UTF-8`. Pour une liste des encodages standard, consultez la section Encodages standard dans la documentation Python.

AWS_CLI_OUTPUT_ENCODING

Spécifie le codage utilisé pour la sortie du AWS CLI. Par défaut, l'encodage correspond à vos paramètres régionaux. Par exemple, si vous utilisez Windows avec l'encodage `CP1252` par défaut, le paramètre `AWS_CLI_OUTPUT_ENCODING=UTF-8` définit l'interface de ligne de commande de sorte que celle-ci ouvre les fichiers texte au format `UTF-8`.

AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS

Si les compartiments source et de destination sont identiques lors de l'utilisation de la `s3 mv` commande, le fichier source ou l'objet peut être déplacé sur lui-même, ce qui peut entraîner la suppression accidentelle de votre fichier ou objet source. La variable et l'`--validate-same-s3-paths`option d'`AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATH`Environnement indiquent s'il faut valider votre point d'accès ARNs ou ses alias dans votre source ou votre destination URIs Amazon S3.

> **Note**
>
> La validation du chemin pour `s3 mv` nécessite des appels d'API supplémentaires.

AWS_CONFIG_FILE

Spécifie l'emplacement du fichier AWS CLI utilisé pour stocker les profils de configuration. Le chemin par défaut est `~/.aws/config`.

Vous ne pouvez pas spécifier cette valeur dans un paramètre de profil nommé ou à l'aide d'un paramètre de ligne de commande.

`AWS_DATA_PATH`

Une liste de répertoires supplémentaires à vérifier en dehors du chemin de recherche intégré ou `~/.aws/models` lors du chargement de AWS CLI données. La définition de cette variable d'environnement indique les répertoires supplémentaires à vérifier avant de revenir au chemin de recherche intégré. Les entrées multiples doivent être séparées par le `os.pathsep` caractère, qui se trouve `:` sous Linux ou macOS et `;` sous Windows.

`AWS_DEFAULT_OUTPUT`

Spécifie le format de sortie à utiliser.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `output`. Vous pouvez remplacer cette variable d'environnement à l'aide du paramètre de ligne de commande `--output`.

`AWS_DEFAULT_REGION`

`Default region name`Identifie la AWS région à laquelle vous souhaitez envoyer vos demandes par défaut aux serveurs. Il s'agit généralement de la région la plus proche de vous, mais il peut s'agir de n'importe quelle autre région. Par exemple, vous pouvez taper `us-west-2` pour utiliser US West (Oregon). Il s'agit de la région à laquelle toutes les requêtes ultérieures sont envoyées, à moins que vous ne spécifiiez une autre option dans une commande individuelle.

Ce paramètre est AWS CLI réservé aux utilisateurs et est conservé pour des raisons de rétrocompatibilité. Il est suggéré d'utiliser plutôt la variable d'`AWS_REGION`environnement compatible avec le AWS SDK.

> **Note**
>
> Vous devez spécifier une AWS région lorsque vous utilisez le AWS CLI, soit explicitement, soit en définissant une région par défaut. Pour connaître la liste des régions disponibles, consultez Régions et points de terminaison. Les désignateurs de région utilisés par le AWS CLI sont les mêmes noms que ceux que vous voyez dans les points de terminaison AWS Management Console URLs de service.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `region`. Vous pouvez remplacer cette variable d'environnement en utilisant le paramètre de ligne de `--region` commande et la variable d'`AWS_REGION`environnement compatible avec le AWS SDK.

`AWS_EC2_METADATA_DISABLED`

Désactive l'utilisation du service de métadonnées d' EC2 instance Amazon (IMDS).

Si ce paramètre est défini sur true, les informations d'identification de l'utilisateur ou la configuration (comme la région) ne sont pas demandées à l'IMDS.

`AWS_ENDPOINT_URL`

Spécifie le point de terminaison utilisé pour toutes les demandes de service. Pour de plus amples informations, veuillez consulter the section called “Définir un point de terminaison global pour tous Services AWS”.

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called “Configuration du point de terminaison et priorité des paramètres”.

`AWS_ENDPOINT_URL_<SERVICE>`

Spécifie un point de terminaison personnalisé utilisé pour un service spécifique, où `<SERVICE>` il est remplacé par l' Service AWS identifiant. Par exemple, Amazon DynamoDB possède un `serviceId` de `DynamoDB`. Pour ce service, la variable d'environnement de l'URL du point de terminaison est`AWS_ENDPOINT_URL_DYNAMODB`.

Pour obtenir la liste de toutes les variables d'environnement spécifiques au service, consultez. Liste des identifiants spécifiques au service

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un

ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`

Si cette option est activée, toutes les configurations de point de terminaison personnalisées sont AWS CLI ignorées. Les valeurs valides sont **`true`** et **`false`**. Pour de plus amples informations, veuillez consulter the section called "Définir un point de terminaison global pour tous Services AWS".

Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. Pour la priorité des points de terminaison, voirthe section called "Configuration du point de terminaison et priorité des paramètres".

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

`AWS_MAX_ATTEMPTS`

Spécifie la valeur maximale de tentatives utilisées par le gestionnaire de AWS CLI nouvelles tentatives, l'appel initial étant pris en compte dans la valeur que vous fournissez. Pour plus d'informations sur les nouvelles tentatives, consultez la section AWS CLI réessaie dans le AWS CLI.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre `max_attempts` des profils.

AWS_METADATA_SERVICE_NUM_ATTEMPTS

Lorsque vous tentez de récupérer des informations d'identification sur une EC2 instance Amazon configurée avec un rôle IAM, le système AWS CLI tente de récupérer les informations d'identification une seule fois auprès du service de métadonnées de l'instance avant de s'arrêter. Si vous savez que vos commandes seront exécutées sur une EC2 instance Amazon, vous pouvez augmenter cette valeur pour AWS CLI réessayer plusieurs fois avant d'abandonner.

AWS_METADATA_SERVICE_TIMEOUT

Le nombre de secondes avant l'expiration d'une connexion au service de métadonnées d'instance. Lorsque vous tentez de récupérer des informations d'identification sur une EC2 instance Amazon configurée avec un rôle IAM, la connexion au service de métadonnées de l'instance expire au bout d'une seconde par défaut. Si vous savez que vous exécutez une EC2 instance Amazon avec un rôle IAM configuré, vous pouvez augmenter cette valeur si nécessaire.

AWS_PAGER

Spécifie le programme de pager utilisé pour la sortie. Par défaut, AWS CLI la version 2 renvoie toutes les sorties via le programme de téléavertisseur par défaut de votre système d'exploitation.

Pour désactiver toute utilisation d'un programme de pagination externe, définissez la variable sur une chaîne vide.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `cli_pager`.

AWS_PROFILE

Spécifie le nom du AWS CLI profil avec les informations d'identification et les options à utiliser. Il peut s'agir du nom d'un profil stocké dans un fichier `credentials` ou `config`, ou de la valeur `default` pour utiliser le profil par défaut.

Si elle est définie, cette variable d'environnement remplace le comportement d'utilisation du profil nommé `[default]` dans le fichier de configuration. Vous pouvez remplacer cette variable d'environnement à l'aide du paramètre de ligne de commande `--profile`.

AWS_REGION

Variable d'environnement compatible avec le AWS SDK qui indique la AWS région à laquelle envoyer la demande.

Si elle est définie, cette variable d'environnement remplace les valeurs de la variable d'environnement `AWS_DEFAULT_REGION` et du paramètre `region` de profil. Vous pouvez

remplacer cette variable d'environnement à l'aide du paramètre de ligne de commande `--region`.

`AWS_REQUEST_CHECKSUM_CALCULATION`

Spécifie le moment où une somme de contrôle est calculée pour les charges utiles des demandes et propose les options suivantes :

- `when_supported`— (Par défaut) La somme de contrôle de la charge utile de la demande est calculée lorsqu'une opération spécifie un algorithme de somme de contrôle dans son modèle de service ou nécessite des sommes de contrôle des demandes.
- `when_required`— La somme de contrôle de la charge utile de la demande est calculée lorsqu'une opération nécessite des sommes de contrôle de demande ou lorsqu'un utilisateur fournit une somme de contrôle `requestAlgorithmMember` modélisée par le. Service AWS

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre `request_checksum_calculation`des profils.

`AWS_RESPONSE_CHECKSUM_VALIDATION`

Spécifie à quel moment la validation de la somme de contrôle est effectuée pour les charges utiles de réponse et propose les options suivantes :

- `when_supported`— (Par défaut) La validation de la somme de contrôle de la charge utile de réponse est effectuée lorsqu'une opération spécifie un algorithme de réponse dans son modèle de service qu'elle prend en charge. AWS CLI
- `when_required`— La validation de la somme de contrôle de la charge utile de réponse est effectuée lorsqu'une opération spécifie un algorithme de réponse dans son modèle de service qu'elle prend en AWS CLI charge, et que vous définissez le modèle sur lequel il est modélisé `ENABLED` dans l'`requestValidationModeMember`entrée de l'opération.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre `response_checksum_validation`des profils.

`AWS_RETRY_MODE`

Spécifie le mode de nouvelle tentative AWS CLI utilisé. Trois modes de nouvelle tentative sont disponibles : `standard` (par défaut), `legacy` et`adaptive`. Pour plus d'informations sur les nouvelles tentatives, consultez la section AWS CLI réessaie dans le AWS CLI.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre `retry_mode` des profils.

`AWS_ROLE_ARN`

Spécifie le nom de ressource Amazon (ARN) d'un rôle IAM auprès d'un fournisseur d'identité Web que vous souhaitez utiliser pour exécuter les AWS CLI commandes.

Utilisé avec les variables `AWS_WEB_IDENTITY_TOKEN_FILE` d'`AWS_ROLE_SESSION_NAME`environnement et.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `role_arn`. Vous ne pouvez pas spécifier un nom de session de rôle en tant que paramètre de ligne de commande.

> **Note**
>
> Cette variable d'environnement s'applique uniquement à un rôle assumé auprès d'un fournisseur d'identité Web ; elle ne s'applique pas à la configuration générale du fournisseur de rôles d'emprunt.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called “Utilisation d'un rôle à l'aide d'une identité web”.

`AWS_ROLE_SESSION_NAME`

Spécifie le nom à attacher à la session de rôle. Cette valeur est fournie au `RoleSessionName` paramètre lorsqu'il AWS CLI appelle l'`AssumeRole`opération et fait partie du rôle assumé par l'utilisateur ARN : `arn:aws:sts::`*`123456789012`*`:assumed-role/`*`role_name`*`/`*`role_session_name`*. Ce paramètre est facultatif. Si vous ne spécifiez pas cette valeur, un nom de session est généré automatiquement. Ce nom apparaît dans AWS CloudTrail les journaux des entrées associées à cette session.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `role_session_name`.

Utilisé avec les variables `AWS_ROLE_ARN` d'`AWS_WEB_IDENTITY_TOKEN_FILE`environnement et.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called “Utilisation d'un rôle à l'aide d'une identité web”.

> **Note**
>
> Cette variable d'environnement s'applique uniquement à un rôle assumé auprès d'un fournisseur d'identité Web ; elle ne s'applique pas à la configuration générale du fournisseur de rôles d'emprunt.

`AWS_SDK_UA_APP_ID`

Un seul Compte AWS peut être utilisé par plusieurs applications clients pour passer des appels à Services AWS. L'ID d'application identifie l'application source qui a effectué un ensemble d'appels à l'aide d'un Service AWS. AWS SDKs et les services n'utilisent ni n'interprètent cette valeur autrement que pour la réintégrer dans les communications avec les clients. Par exemple, cette valeur peut être incluse dans les e-mails opérationnels pour identifier de manière unique laquelle de vos applications est associée à la notification.

Par défaut, il n'y a aucune valeur.

L'ID de l'application est une chaîne d'une longueur maximale de 50 caractères. Les lettres, les chiffres et les caractères spéciaux suivants sont autorisés :

```
! $ % & * + - . , ^ _ ` | ~
```

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `sdk_ua_app_id`. Vous ne pouvez pas spécifier l'ID de l'application comme option de ligne de commande.

`AWS_SECRET_ACCESS_KEY`

Indique la clé secrète associée à la clé d'accès. Il s'agit du « mot de passe » de la clé d'accès.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `aws_secret_access_key`. Vous ne pouvez pas spécifier l'ID de la clé d'accès secrète en tant qu'option de ligne de commande.

`AWS_SESSION_TOKEN`

Spécifie la valeur du jeton de session requise si vous utilisez des informations d'identification de sécurité temporaires que vous avez extraites directement des opérations AWS STS . Pour plus d'informations, consultez la section Sortie de la commande assume-role dans la référence des AWS CLI commandes.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `aws_session_token`.

`AWS_SHARED_CREDENTIALS_FILE`

Spécifie l'emplacement du fichier AWS CLI utilisé pour stocker les clés d'accès. Le chemin par défaut est `~/.aws/credentials`.

Vous ne pouvez pas spécifier cette valeur dans un paramètre de profil nommé ou à l'aide d'un paramètre de ligne de commande.

`AWS_SIGV4A_SIGNING_REGION_SET`

Spécifie les régions à utiliser lors de la signature avec SigV4a à l'aide d'une liste séparée par des virgules. Si cette variable n'est pas définie, elle AWS CLI utilise la valeur par défaut utilisée par le Service AWS. S'il n'y Service AWS a pas de valeur par défaut, la signature de la demande devient valide dans toutes les régions en utilisant une valeur de`*`.

Pour plus d'informations sur SIGv4a, voir AWS Signature version 4 pour les demandes d'API dans le guide de l'utilisateur IAM

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `sigv4a_signing_region_set`.

`AWS_USE_DUALSTACK_ENDPOINT`

Permet d'utiliser des points de terminaison à double pile pour envoyer AWS des demandes. Pour en savoir plus sur les points de terminaison à double pile, qui prennent en charge à la fois le IPv6 trafic IPv4 et le trafic, consultez la section Utilisation des points de terminaison à double pile Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service. Des points de terminaison à double pile sont disponibles pour certains services dans certaines régions. Si aucun point de terminaison à double pile n'existe pour le service ou Région AWS si la demande échoue. Par défaut, l'option est désactivée. Pour de plus amples informations, veuillez consulter the section called “Conçu pour utiliser des points de terminaison à double pile pour tous Services AWS”.

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un

ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

AWS_USE_FIPS_ENDPOINT

Certains AWS services proposent des terminaux compatibles avec la norme fédérale de traitement de l'information (FIPS) 140-2 dans certains cas. Régions AWS Lorsque le AWS service prend en charge le protocole FIPS, ce paramètre indique le point de terminaison FIPS à utiliser AWS CLI . Contrairement aux AWS terminaux standard, les points de terminaison FIPS utilisent une bibliothèque logicielle TLS conforme à la norme FIPS 140-2. Ces points de terminaison peuvent être requis par les entreprises qui interagissent avec le gouvernement des États-Unis. Pour plus d'informations, voir,the section called "Paramétré pour utiliser les FIPs points de terminaison pour tous Services AWS".

Si ce paramètre est activé, mais qu'il n'existe pas de point de terminaison FIPS pour le service dans votre ordinateur Région AWS, la AWS commande risque d'échouer. Dans ce cas, spécifiez manuellement le point de terminaison à utiliser dans la commande à l'aide de l'`--endpoint-url`option ou utilisez des points de terminaison spécifiques au service.

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

AWS_WEB_IDENTITY_TOKEN_FILE

Spécifie le chemin d'accès à un fichier contenant un jeton d'accès OAuth 2.0 ou un jeton d'identification OpenID Connect fourni par un fournisseur d'identité. L' AWS CLI charge le contenu de ce fichier et le transmet en tant qu'argument `WebIdentityToken` à l'opération `AssumeRoleWithWebIdentity`.

Utilisé avec les variables `AWS_ROLE_ARN` d'`AWS_ROLE_SESSION_NAME`environnement et.

Si elle est définie, cette variable d'environnement remplace la valeur du paramètre de profil `web_identity_token_file`.

Pour plus d'informations sur l'utilisation des identités Web, consultezthe section called “Utilisation d'un rôle à l'aide d'une identité web”.

> **Note**
>
> Cette variable d'environnement s'applique uniquement à un rôle assumé auprès d'un fournisseur d'identité Web ; elle ne s'applique pas à la configuration générale du fournisseur de rôles d'emprunt.

Options de ligne de commande dans AWS CLI

Dans le AWS CLI, les options de ligne de commande sont des paramètres globaux que vous pouvez utiliser pour remplacer les paramètres de configuration par défaut, tout paramètre de profil correspondant ou le paramètre de variable d'environnement pour cette commande unique. Vous ne pouvez pas utiliser les options de ligne de commande pour spécifier directement des informations d'identification, bien que vous puissiez spécifier le profil à utiliser.

Rubriques

- Comment utiliser les options de ligne de commande
- AWS CLI options de ligne de commande globales prises en charge
- Utilisations courantes des options de ligne de commande

Comment utiliser les options de ligne de commande

La plupart des options de ligne de commande sont de simples chaînes, comme le nom du profil `profile1` dans l'exemple suivant :

```
$ aws s3 ls --profile profile1
amzn-s3-demo-bucket1
amzn-s3-demo-bucket2
...
```

Chaque option qui prend un argument nécessite un espace ou un signe égal (=) qui sépare l'argument du nom de l'option. Si la valeur d'argument est une chaîne qui contient un espace, vous devez placer l'argument entre guillemets. Pour plus de détails sur les types d'arguments et le formatage des paramètres, consultezSpécifier les valeurs des paramètres dans le AWS CLI.

AWS CLI options de ligne de commande globales prises en charge

Dans le, AWS CLI vous pouvez utiliser les options de ligne de commande suivantes pour remplacer les paramètres de configuration par défaut, tout paramètre de profil correspondant ou le paramètre de variable d'environnement pour cette commande unique.

--ca-bundle ***<string>***

Spécifie l'ensemble de certificats d'autorité de certification (CA) à utiliser lors de la vérification des certificats SSL.

Si elle est définie, cette option remplace la valeur du paramètre de profil `ca_bundle` et de la variable d'`AWS_CA_BUNDLE`environnement.

--cli-auto-prompt

Active le mode d'invite automatique pour une seule commande. Comme le montrent les exemples suivants, vous pouvez le spécifier à tout moment.

```
$ aws --cli-auto-prompt
$ aws dynamodb --cli-auto-prompt
$ aws dynamodb describe-table --cli-auto-prompt
```

Cette option remplace la variable d'`aws_cli_auto_prompt`environnement et le paramètre de `cli_auto_prompt` profil.

Pour plus d'informations sur la fonctionnalité d'invite automatique de la AWS CLI version 2, consultezActivation et utilisation des invites de commande dans le AWS CLI.

--cli-binary-format

Spécifie la manière dont la AWS CLI version 2 interprète les paramètres d'entrée binaires. Il peut avoir l'une des valeurs suivantes :

- base64 — Il s'agit de la valeur par défaut. Un paramètre d'entrée qui est saisi comme un objet BLOB accepte une chaîne codée en base64. Pour transmettre le contenu binaire réel, placez le contenu dans un fichier et indiquez le chemin d'accès et le nom du fichier avec le préfixe

`fileb://` comme valeur du paramètre. Pour transmettre du texte codé en base64 contenu dans un fichier, indiquez le chemin d'accès et le nom du fichier avec le préfixe `file://` comme valeur du paramètre.

- raw-in-base64-out — Valeur par défaut pour la AWS CLI version 1. Si la valeur du paramètre est`raw-in-base64-out`, les fichiers référencés à l'aide du `file://` préfixe sont lus sous forme de texte, puis les AWS CLI tentatives de codage en binaire sont tentées.

Cela remplace le paramètre de configuration `cli_binary_format` du fichier.

```
$ aws lambda invoke \
    --cli-binary-format raw-in-base64-out \
    --function-name my-function \
    --invocation-type Event \
    --payload '{ "name": "Bob" }' \
    response.json
```

Si vous référencez une valeur binaire dans un fichier à l'aide de la notation `fileb://` préfixe, le fichier s'attend AWS CLI toujours à ce que le fichier contienne du contenu binaire brut et ne tente pas de convertir la valeur.

Si vous référencez une valeur binaire dans un fichier à l'aide de la notation `file://` préfixe, AWS CLI le fichier est géré conformément au `cli_binary_format` paramètre actuel.
Si la valeur de ce paramètre est `base64` (la valeur par défaut lorsqu'elle n'est pas définie explicitement), le fichier AWS CLI s'attend à ce que le fichier contienne du texte codé en base64.
Si la valeur de ce paramètre est`raw-in-base64-out`, le fichier AWS CLI s'attend à ce que le fichier contienne du contenu binaire brut.

--cli-connect-timeout ***<integer>***

Spécifie temps de connexion maximal du socket en secondes. Si la valeur est définie sur 0, la connexion du socket attend indéfiniment (est bloquante) sans délai d'expiration. La valeur par défaut est de 60 secondes.

--cli-read-timeout ***<integer>***

Spécifie temps de lecture maximal du socket en secondes. Si la valeur est définie sur 0, la lecture du socket attend indéfiniment sans délai d'expiration. La valeur par défaut est de 60 secondes.

--couleur ***<string>***

Spécifie la prise en charge de la sortie couleur. Les valeurs valides sont on, `off` et `auto`. La valeur par défaut est `auto`.

--debug

Commutateur booléen qui active la journalisation du débogage. AWS CLI Par défaut, fournit des informations nettoyées concernant les succès ou les échecs concernant les résultats des commandes dans la sortie de commande. L'`--debug`option fournit les journaux Python complets. Cela inclut des informations `stderr` de diagnostic supplémentaires sur le fonctionnement de la commande qui peuvent être utiles pour résoudre les problèmes liés aux résultats inattendus d'une commande. Pour consulter facilement les journaux de débogage, nous vous suggérons de les envoyer dans un fichier afin de rechercher plus facilement les informations. Vous pouvez le faire en utilisant l'une des méthodes suivantes.

Pour envoyer uniquement les informations `stderr` de diagnostic, ajoutez `2> debug.txt` le nom que vous souhaitez utiliser pour votre fichier de débogage : `debug.txt`

```
$ aws servicename commandname options --debug 2> debug.txt
```

Pour envoyer à la fois les informations `stderr` de sortie et de diagnostic, `&> debug.txt` ajoutez `debug.txt` le nom que vous souhaitez utiliser pour votre fichier de débogage :

```
$ aws servicename commandname options --debug &> debug.txt
```

--endpoint-url ***<string>***

Spécifie l'URL à laquelle envoyer la requête. Pour la plupart des commandes, l'URL est AWS CLI automatiquement déterminée en fonction du service sélectionné et de la AWS région spécifiée. Toutefois, certaines commandes nécessitent que vous spécifiez une URL propre au compte. Vous pouvez également configurer certains AWS services pour héberger un point de terminaison directement dans votre VPC privé, qui devra peut-être ensuite être spécifié.

L'exemple de commande suivant utilise une URL de point de terminaison Amazon S3 personnalisée.

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

> **Priorité des points de terminaison**
>
> Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en

tant que paramètre. AWS CLI vérifie ces paramètres de point de terminaison dans un ordre particulier et utilise le paramètre de point de terminaison ayant la priorité la plus élevée. Pour la liste des priorités des points de terminaison, consultezthe section called "Configuration du point de terminaison et priorité des paramètres".

--no-cli-auto-prompt

Désactive le mode d'invite automatique pour une seule commande.

```
$ aws dynamodb describe-table --table-name Table1 --no-cli-auto-prompt
```

Cette option remplace la variable d'`aws_cli_auto_prompt`environnement et le paramètre de `cli_auto_prompt` profil.

Pour plus d'informations sur la fonctionnalité d'invite automatique de la AWS CLI version 2, consultezActivation et utilisation des invites de commande dans le AWS CLI.

--no-cli-pager

Commutateur booléen qui désactive l'utilisation d'un pager pour la sortie de la commande.

--no-paginate

Commutateur booléen qui désactive les multiples appels effectués automatiquement pour recevoir tous les résultats de commande, AWS CLI ce qui crée une pagination de la sortie. Cela signifie que seule la première page de votre sortie est affichée.

--no-sign-request

Commutateur booléen qui désactive la signature des requêtes HTTP vers le point de terminaison du AWS service. Cela empêche le chargement d'informations d'identification.

--no-verify-ssl

Par défaut, le protocole SSL est AWS CLI utilisé pour communiquer avec AWS les services. Pour chaque connexion et appel SSL, les certificats SSL sont AWS CLI vérifiés. L'utilisation de cette option remplace le comportement par défaut de vérification des certificats SSL.

> ⚠ Warning
>
> Cette option n'est pas une bonne pratique. Si vous l'utilisez`--no-verify-ssl`, votre trafic entre votre client et les AWS services n'est plus sécurisé. Cela signifie que votre

trafic représente un risque de sécurité et est vulnérable aux man-in-the-middle exploits. Si vous rencontrez des problèmes avec les certificats, il est préférable de les résoudre. Pour les étapes de résolution des problèmes liés aux certificats, consultezthe section called "Erreurs de certificat SSL".

--sortie ***<string>***

Spécifie le format de sortie à utiliser pour cette commande. Vous pouvez spécifier les valeurs suivantes :

- **`json`**— La sortie est formatée sous forme de chaîne JSON.
- **`yaml`**— La sortie est formatée sous forme de chaîne YAML.
- **`yaml-stream`**— La sortie est diffusée et formatée sous forme de chaîne YAML. Le streaming permet de traiter plus rapidement de gros types de données.
- **`text`**— La sortie est formatée sous la forme de plusieurs lignes de valeurs de chaîne séparées par des tabulations. Cela peut être utile pour transmettre le résultat à un processeur de texte, comme `grep`ed, ou`awk`.
- **`table`**— La sortie est formatée sous forme de tableau en utilisant les caractères +|- pour former les bordures des cellules. La présentation des informations est dans un format beaucoup plus lisible par l'utilisateur que les autres, mais peu pratique du point de vue programmation.

--profil ***<string>***

Spécifie le profil nommé à utiliser pour cette commande. Pour configurer des profils nommés supplémentaires, vous pouvez utiliser la commande `aws configure` avec l'option `--profile`.

```
$ aws configure --profile <profilename>
```

--requête ***<string>***

Spécifie une JMESPath requête à utiliser pour filtrer les données de réponse. Pour de plus amples informations, veuillez consulter Filtrage de la sortie dans le AWS CLI.

--région ***<string>***

Spécifie AWS la région à laquelle envoyer la AWS demande de cette commande. Pour obtenir la liste de toutes les régions que vous pouvez spécifier, consultez Régions et points de terminaison AWS dans le document Référence générale d'Amazon Web Services.

--Version

Commutateur booléen qui affiche la version actuelle du AWS CLI programme en cours d'exécution.

Utilisations courantes des options de ligne de commande

Ces options de ligne de commande sont couramment utilisées pour vérifier vos ressources dans plusieurs régions AWS et modifier le format de sortie pour l'éligibilité ou la facilité d'utilisation lors de l'utilisation de scripts. Dans les exemples suivants, nous exécutons la commande describe-instances sur chaque région jusqu'à ce que nous trouvions la région dans laquelle se trouve notre instance.

```
$ aws ec2 describe-instances --output table --region us-west-1
------------------
|DescribeInstances|
+-----------------+
$ aws ec2 describe-instances --output table --region us-west-2
-------------------------------------------------------------------------------
|                            DescribeInstances                                |
+-----------------------------------------------------------------------------+
||                              Reservations                                 ||
|+-----------------------------------+-----------------------------------+|
||  OwnerId                          |  012345678901                     ||
||  ReservationId                    |  r-abcdefgh                       ||
|+-----------------------------------+-----------------------------------+|
|||                               Instances                               |||
||+-------------------------+-----------------------------------------------+||
|||  AmiLaunchIndex         |  0                                            |||
|||  Architecture           |  x86_64                                       |||
...
```

Configuration de l'exécution des commandes dans AWS CLI

Le AWS Command Line Interface (AWS CLI) inclut une fonction de complétion de commande compatible avec bash qui vous permet d'utiliser la touche Tab pour terminer une commande partiellement saisie. Sur la plupart des systèmes, vous devez le configurer manuellement.

Pour plus d'informations sur la fonctionnalité d'invite automatique de la AWS CLI version 2 à la place, voirActivation et utilisation des invites de commande dans le AWS CLI.

Rubriques

- Comment ça marche
- Configuration de l'exécution des commandes sous Linux ou macOS
- Configuration de l'exécution des commandes sous Windows

Comment ça marche

Lorsque vous saisissez partiellement une commande, un paramètre ou une option, la fonction de complétion de commande complète automatiquement votre commande ou affiche une liste de commandes suggérées. Pour demander la fin d'une commande, vous devez entrer partiellement une commande et appuyer sur la touche de fin, qui se trouve généralement *Tab* dans la plupart des interpréteurs de commandes.

Les exemples suivants montrent différentes façons d'utiliser la complétion d'une commande :

- Entrez partiellement une commande et appuyez sur cette touche *Tab* pour afficher une liste de commandes suggérées.

```
$ aws dynamodb dTAB
delete-backup                       describe-global-table
delete-item                         describe-global-table-settings
delete-table                        describe-limits
describe-backup                     describe-table
describe-continuous-backups         describe-table-replica-auto-scaling
describe-contributor-insights       describe-time-to-live
describe-endpoints
```

- Entrez partiellement un paramètre et appuyez sur cette touche *Tab* pour afficher une liste de paramètres suggérée.

```
$ aws dynamodb delete-table --TAB
--ca-bundle              --endpoint-url           --profile
--cli-connect-timeout    --generate-cli-skeleton  --query
--cli-input-json         --no-paginate            --region
--cli-read-timeout       --no-sign-request        --table-name
--color                  --no-verify-ssl          --version
--debug                  --output
```

- Entrez un paramètre et appuyez dessus *Tab* pour afficher une liste suggérée de valeurs de ressources. Cette fonctionnalité n'est disponible que dans la AWS CLI version 2.

```
$ aws dynamodb delete-table --table-name TAB
Table 1                  Table 2                  Table 3
```

Configuration de l'exécution des commandes sous Linux ou macOS

Pour configurer l'exécution des commandes sous Linux ou macOS, vous devez connaître le nom du shell que vous utilisez et l'emplacement du `aws_completer` script.

> **Note**
>
> L'exécution des commandes est automatiquement configurée et activée par défaut sur les EC2 instances Amazon qui exécutent Amazon Linux.

Rubriques

- Vérifiez que le dossier du compléteur se trouve dans votre chemin
- Activer la saisie automatique de commandes
- Vérifier l'exécution de la commande

Vérifiez que le dossier du compléteur se trouve dans votre chemin

Pour que le AWS completer fonctionne correctement, il `aws_completer` doit se trouver sur le chemin de votre shell. La `which` commande peut vérifier si le compléteur se trouve sur votre chemin.

```
$ which aws_completer
/usr/local/bin/aws_completer
```

Si la commande which ne trouve pas le compléteur, suivez les étapes ci-dessous pour ajouter le dossier du compléteur à votre chemin.

Étape 1 : Localiser le AWS compléteur

L'emplacement du AWS compléteur peut varier en fonction de la méthode d'installation utilisée.

- Gestionnaire de packages : les programmes tels que `pip` `yumbrew`, et `apt-get` installent généralement le AWS compléteur (ou un lien symbolique vers celui-ci) sur un emplacement de chemin standard.

  - Si vous l'avez utilisé `pip` sans le `--user` paramètre, le chemin par défaut est`/usr/local/bin/aws_completer`.
  - Si vous l'avez utilisé `pip` avec le `--user` paramètre, le chemin par défaut est`/home/username/.local/bin/aws_completer`.
- Programme d'installation groupé : si vous avez utilisé le programme d'installation groupé, le chemin par défaut est. `/usr/local/bin/aws_completer`

Si tout échoue, vous pouvez utiliser la `find` commande pour rechercher le AWS compléteur dans votre système de fichiers.

```
$ find / -name aws_completer
/usr/local/bin/aws_completer
```

Étape 2 : Identifiez votre coque

Pour identifier le shell que vous utilisez, vous pouvez utiliser l'une des commandes suivantes.

- echo $SHELL — Affiche le nom du fichier programme du shell. Il correspond généralement au nom du shell que vous utilisez, sauf si vous avez lancé un autre shell après la connexion.

  ```
  $ echo $SHELL
  /bin/bash
  ```

- ps — Affiche les processus en cours pour l'utilisateur actuel. L'un d'eux est la coquille.

  ```
  $ ps
    PID TTY          TIME CMD
   2148 pts/1    00:00:00 bash
   8756 pts/1    00:00:00 ps
  ```

Étape 3 : Ajoutez le compléteur à votre parcours

1. Recherchez le script de profil de votre shell dans votre dossier utilisateur.

   ```
   $ ls -a ~/
   .  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
   ```

   - Bash — `.bash_profile`.`profile`, ou `.bash_login`

- Zash — `.zshrc`
- Tcsh— `.tcshrc.cshrc`, ou `.login`

2. Ajoutez une commande d'exportation à la fin de votre script de profil similaire à l'exemple suivant. Remplacez */usr/local/bin/* par le nom du dossier que vous avez découvert dans la section précédente.

```
export PATH=/usr/local/bin/:$PATH
```

3. Rechargez le profil dans la session en cours pour appliquer ces modifications. Remplacez *.bash_profile* par le nom du script de shell que vous avez découvert dans la première section.

```
$ source ~/.bash_profile
```

Activer la saisie automatique de commandes

Après avoir confirmé que le compléteur se trouve sur votre chemin, activez l'exécution des commandes en exécutant la commande appropriée pour le shell que vous utilisez. Vous pouvez ajouter la commande au profil de votre shell pour l'exécuter chaque fois que vous ouvrez un nouveau shell. Dans chaque commande, remplacez le */usr/local/bin/* chemin par celui trouvé sur votre système dansVérifiez que le dossier du compléteur se trouve dans votre chemin.

- **`bash`**— Utilisez la commande intégrée`complete`.

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

  Ajoutez la commande précédente `~/.bashrc` pour l'exécuter chaque fois que vous ouvrez un nouveau shell. Votre `~/.bash_profile` doit avoir pour source `~/.bashrc` pour garantir que la commande est également exécutée dans les shells de connexion.

- **`zsh`**— Pour exécuter l'exécution de la commande, vous devez l'exécuter `bashcompinit` en ajoutant la ligne de chargement automatique suivante à la fin de votre script de `~/.zshrc` profil.

```
$ autoload bashcompinit && bashcompinit
$ autoload -Uz compinit && compinit
```

  Pour activer la complétion d'une commande, utilisez la commande intégrée `complete`.

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

Ajoutez les commandes précédentes `~/.zshrc` pour l'exécuter chaque fois que vous ouvrez un nouveau shell.

- **`tcsh`**— Complète pour `tcsh` prend un type de mot et un modèle pour définir le comportement de complétion.

```
> complete aws 'p/*/`aws_completer`/'
```

Ajoutez la commande précédente `~/.tcshrc` pour l'exécuter chaque fois que vous ouvrez un nouveau shell.

Une fois que vous avez activé l'exécution des commandes, cela Vérifier l'exécution de la commande fonctionne.

Vérifier l'exécution de la commande

Après avoir activé l'exécution des commandes, rechargez votre shell, entrez une commande partielle et appuyez sur Tab pour voir les commandes disponibles.

```
$ aws sTAB
s3              ses             sqs             sts             swf
s3api           sns             storagegateway  support
```

Configuration de l'exécution des commandes sous Windows

> **Note**
>
> Pour plus d'informations sur la façon dont PowerShell ils sont complétés, y compris leurs différentes clés d'achèvement, voir about_TAB_Expansion dans les Microsoft Docs. PowerShell

Pour activer l'exécution PowerShell des commandes sous Windows, procédez comme suit dans PowerShell.

1. Ouvrez votre fichier `$PROFILE` à l'aide de la commande suivante.

```
PS C:\> Notepad $PROFILE
```

Si vous n'en avez pas$PROFILE, créez un profil utilisateur à l'aide de la commande suivante.

```
PS C:\> if (!(Test-Path -Path $PROFILE ))
{ New-Item -Type File -Path $PROFILE -Force }
```

Pour plus d'informations sur PowerShell les profils, consultez Comment utiliser les profils dans Windows PowerShell ISE sur le site Web de Microsoft Docs.

2. Pour activer l'exécution des commandes, ajoutez le bloc de code suivant à votre profil, enregistrez puis fermez le fichier.

```
Register-ArgumentCompleter -Native -CommandName aws -ScriptBlock {
    param($commandName, $wordToComplete, $cursorPosition)
        $env:COMP_LINE=$wordToComplete
        if ($env:COMP_LINE.Length -lt $cursorPosition){
            $env:COMP_LINE=$env:COMP_LINE + " "
        }
        $env:COMP_POINT=$cursorPosition
        aws_completer.exe | ForEach-Object {
            [System.Management.Automation.CompletionResult]::new($_, $_,
 'ParameterValue', $_)
        }
        Remove-Item Env:\COMP_LINE
        Remove-Item Env:\COMP_POINT
}
```

3. Après avoir activé l'exécution des commandes, rechargez votre shell, entrez une commande partielle et appuyez sur Tab pour parcourir les commandes disponibles.

```
$ aws sTab
```

```
$ aws s3
```

Pour voir toutes les commandes disponibles disponibles jusqu'à ce que vous ayez terminé, entrez une commande partielle et appuyez sur Ctrl + Espace.

```
$ aws sCtrl + Space
```

```
s3              ses             sqs             sts             swf
s3api           sns             storagegateway  support
```

AWS CLI réessaie dans le AWS CLI

Cette rubrique décrit comment les appels aux AWS services AWS CLI peuvent échouer en raison de problèmes inattendus. Ces problèmes peuvent se produire côté serveur ou échouer en raison de la limitation du débit du AWS service que vous essayez d'appeler. Ces types de défaillances ne nécessitent généralement pas de traitement spécial et l'appel est automatiquement refait, souvent après une brève période d'attente. AWS CLI Il fournit de nombreuses fonctionnalités pour aider les clients à réessayer d'appeler les AWS services en cas d'erreurs ou d'exceptions de ce type.

Rubriques

- Modes de nouvelle tentative disponibles
- Configuration d'un mode de nouvelle tentative
- Afficher les journaux des tentatives de nouvelle tentative

Modes de nouvelle tentative disponibles

Vous AWS CLI avez le choix entre plusieurs modes en fonction de votre version :

- Mode de nouvelle tentative standard (par défaut)
- Mode de nouvelle tentative classique
- Mode de nouvelle tentative adaptatif

Mode de nouvelle tentative standard (par défaut)

`Standard`le mode est un ensemble standard de règles de nouvelle tentative AWS SDKs offrant plus de fonctionnalités que l'ancien. Le mode par défaut est le mode par défaut pour AWS CLI. Le mode standard a été créé pour la AWS CLI version 2 et est rétroporté vers la AWS CLI version 1. Les fonctionnalités du mode standard incluent :

- Une valeur par défaut de 2 pour le nombre maximum de tentatives, soit un total de 3 tentatives d'appel. Cette valeur peut être remplacée par le biais du paramètre `max_attempts` de configuration.
- Réessayez pour la liste étendue d'erreurs/exceptions suivante :

- Erreurs/exceptions transitoires
  - `RequestTimeout`
  - `RequestTimeoutException`
  - `PriorRequestNotComplete`
  - `ConnectionError`
  - `HTTPClientError`
- throttling/limit Erreurs et exceptions liées au service :
  - `Throttling`
  - `ThrottlingException`
  - `ThrottledException`
  - `RequestThrottledException`
  - `TooManyRequestsException`
  - `ProvisionedThroughputExceededException`
  - `TransactionInProgressException`
  - `RequestLimitExceeded`
  - `BandwidthLimitExceeded`
  - `LimitExceededException`
  - `RequestThrottled`
  - `SlowDown`
  - `EC2ThrottledException`

- Réessayez de corriger les codes d'erreur transitoires et non descriptifs. Plus précisément, ces codes d'état HTTP : 500, 502, 503, 504.
- Toute nouvelle tentative inclura un recul exponentiel d'un facteur de base de 2 pour une durée maximale de 20 secondes.

Mode de nouvelle tentative classique

`Legacy`le mode utilise un ancien gestionnaire de tentatives dont les fonctionnalités sont limitées, notamment :

- La valeur par défaut est de 4 pour le nombre maximum de tentatives, soit un total de 5 tentatives d'appel. Cette valeur peut être remplacée par le biais du paramètre `max_attempts` de configuration.
- DynamoDB a une valeur par défaut de 9 pour le nombre maximum de tentatives, soit un total de 10 tentatives d'appel. Cette valeur peut être remplacée par le biais du paramètre `max_attempts` de configuration.
- Réessayez pour le nombre limité d'erreurs/exceptions suivant :
  - socket/connection Erreurs générales :
    - `ConnectionError`
    - `ConnectionClosedError`
    - `ReadTimeoutError`
    - `EndpointConnectionError`
  - throttling/limit Erreurs et exceptions liées au service :
    - `Throttling`
    - `ThrottlingException`
    - `ThrottledException`
    - `RequestThrottledException`
    - `ProvisionedThroughputExceededException`
- Réessayez d'utiliser plusieurs codes d'état HTTP, notamment 429, 500, 502, 503, 504 et 509.
- Toute nouvelle tentative comportera un recul exponentiel d'un facteur de base de 2.

Mode de nouvelle tentative adaptatif

> **Warning**
>
> Le mode adaptatif est un mode expérimental susceptible de changer, tant au niveau des fonctionnalités que du comportement.

`Adaptive`le mode nouvelle tentative est un mode de nouvelle tentative expérimental qui inclut toutes les fonctionnalités du mode standard. Outre les fonctionnalités du mode standard, le mode adaptatif introduit également une limitation du débit côté client grâce à l'utilisation d'un bucket de jetons et de variables de limite de débit qui sont mises à jour dynamiquement à chaque nouvelle tentative. Ce

mode offre une flexibilité lors des tentatives côté client qui s'adapte à la réponse de l' error/exception état d'un service. AWS

À chaque nouvelle tentative, le mode adaptatif modifie les variables de limite de débit en fonction de l'erreur, de l'exception ou du code d'état HTTP présenté dans la réponse du service. AWS Ces variables de limite de débit sont ensuite utilisées pour calculer un nouveau débit d'appels pour le client. Chaque réponse HTTP exception/error ou négative (fournie dans la liste ci-dessus) d'un AWS service met à jour les variables de limite de débit au fur et à mesure des nouvelles tentatives jusqu'à ce que le succès soit atteint, que le bucket de jetons soit épuisé ou que la valeur maximale de tentatives configurée soit atteinte.

Configuration d'un mode de nouvelle tentative

AWS CLI Cela inclut une variété de configurations de nouvelle tentative ainsi que des méthodes de configuration à prendre en compte lors de la création de votre objet client.

Méthodes de configuration disponibles

Dans le AWS CLI, les utilisateurs peuvent configurer les nouvelles tentatives de la manière suivante :

- Variables d’environnement
- AWS CLI fichier de configuration

Les utilisateurs peuvent personnaliser les options de nouvelle tentative suivantes :

- Mode de nouvelle tentative - Spécifie le mode de nouvelle tentative utilisé. AWS CLI Comme décrit précédemment, trois modes de nouvelle tentative sont disponibles : ancien, standard et adaptatif. La valeur par défaut pour la version 2 est standard.
- Nombre maximal de tentatives - Spécifie la valeur maximale de tentatives utilisées par le gestionnaire de AWS CLI nouvelles tentatives, l'appel initial étant pris en compte dans la valeur que vous fournissez. La valeur par défaut dépend de votre mode de nouvelle tentative.

Définition d'une configuration de nouvelle tentative dans vos variables d'environnement

Pour définir votre configuration de nouvelle tentative pour le AWS CLI, mettez à jour les variables d'environnement de votre système d'exploitation.

Les variables d'environnement de nouvelle tentative sont les suivantes :

- `AWS_RETRY_MODE`
- `AWS_MAX_ATTEMPTS`

Pour plus d'informations sur les variables d'environnement, consultezConfiguration des variables d'environnement pour AWS CLI.

Définition d'une configuration de nouvelle tentative dans votre fichier AWS de configuration

Pour modifier votre configuration de nouvelle tentative, mettez à jour votre fichier AWS de configuration global. L'emplacement par défaut de votre fichier de configuration est AWS ~/.aws/config.

Voici un exemple de fichier de AWS configuration :

```
[default]
retry_mode = standard
max_attempts = 6
```

Pour plus d'informations sur les fichiers de configuration, consultezParamètres des fichiers de configuration et d'identification dans le AWS CLI.

Afficher les journaux des tentatives de nouvelle tentative

Il AWS CLI utilise la méthodologie de réessai et la journalisation de Boto3. Vous pouvez utiliser `--debug` cette option sur n'importe quelle commande pour recevoir les journaux de débogage. Pour plus d'informations sur l'utilisation de `--debug` cette option, consultezOptions de ligne de commande dans AWS CLI.

Si vous recherchez « réessayer » dans vos journaux de débogage, vous trouverez les informations dont vous avez besoin pour réessayer. Les entrées du journal client pour les nouvelles tentatives dépendent du mode de nouvelle tentative que vous avez activé.

Mode Héritage :

Les messages de nouvelle tentative sont générés par botocore.retryhandler. L'un des trois messages suivants s'affichera :

- `No retry needed`

- `Retry needed, action of: <action_name>`
- `Reached the maximum number of retry attempts: <attempt_number>`

Mode standard ou adaptatif :

Les messages de nouvelle tentative sont générés par botocore.retries.standard. L'un des trois messages suivants s'affichera :

- `No retrying request`
- `Retry needed, retrying request after delay of: <delay_value>`
- `Retry needed but retry quota reached, not retrying request`

Pour le fichier de définition complète des retentatives de botocore, consultez _retry.json dans le référentiel botocore. GitHub

Utilisation d'un proxy HTTP pour AWS CLI

Pour accéder AWS via des serveurs proxy, vous pouvez configurer les variables d'`HTTPS_PROXY`environnement `HTTP_PROXY` et avec les noms de domaine DNS ou les adresses IP et les numéros de port utilisés par vos serveurs proxy.

Rubriques

- Utilisation des exemples
- Authentification auprès d'un proxy
- Utilisation d'un proxy sur les EC2 instances Amazon
- Résolution des problèmes

Utilisation des exemples

> **Note**
>
> Les exemples suivants montrent le nom de la variable d'environnement en majuscules. Toutefois, si vous spécifiez une variable deux fois en utilisant des majuscules différentes, les lettres minuscules ont priorité. Nous vous recommandons de ne définir chaque variable

qu'une seule fois afin d'éviter toute confusion dans le système et tout comportement inattendu.

Les exemples suivants illustrent comment vous pouvez utiliser l'adresse IP explicite de votre proxy ou un nom DNS qui est résolu sur l'adresse IP de votre proxy. Chacun peut être suivi d'un signe deux-points et du numéro de port auquel les requêtes doivent être envoyées.

Linux or macOS

```
$ export HTTP_PROXY=http://10.15.20.25:1234
$ export HTTP_PROXY=http://proxy.example.com:1234
$ export HTTPS_PROXY=http://10.15.20.25:5678
$ export HTTPS_PROXY=http://proxy.example.com:5678
```

Windows Command Prompt

À configurer pour toutes les sessions

```
C:\> setx HTTP_PROXY http://10.15.20.25:1234
C:\> setx HTTP_PROXY http://proxy.example.com:1234
C:\> setx HTTPS_PROXY http://10.15.20.25:5678
C:\> setx HTTPS_PROXY http://proxy.example.com:5678
```

L'utilisation de `setx` pour définir une variable d'environnement modifie la valeur utilisée dans la session d'invite de commande en cours et toutes les sessions d'invite de commande que vous créez après l'exécution de la commande. Cela n'affecte pas les autres shells de commande qui sont déjà en cours d'exécution lorsque vous exécutez la commande.

À configurer pour la session en cours uniquement

L'utilisation de `set` pour définir une variable d'environnement modifie la valeur utilisée jusqu'à la fin de la session d'invite de commande en cours, ou jusqu'à ce que vous définissiez la variable sur une autre valeur.

```
C:\> set HTTP_PROXY=http://10.15.20.25:1234
C:\> set HTTP_PROXY=http://proxy.example.com:1234
C:\> set HTTPS_PROXY=http://10.15.20.25:5678
C:\> set HTTPS_PROXY=http://proxy.example.com:5678
```

Authentification auprès d'un proxy

> **Note**
>
> Ne AWS CLI prend pas en charge les proxys NTLM. Si vous utilisez un proxy de protocole NTLM ou Kerberos, vous pouvez vous connecter via un proxy d'authentification tel que Cntlm.

AWS CLI Supporte l'authentification HTTP Basic. Spécifiez le nom d'utilisateur et le mot de passe dans l'URL du proxy, comme suit.

Linux or macOS

```
$ export HTTP_PROXY=http://username:password@proxy.example.com:1234
$ export HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Windows Command Prompt

À configurer pour toutes les sessions

```
C:\> setx HTTP_PROXY http://username:password@proxy.example.com:1234
C:\> setx HTTPS_PROXY http://username:password@proxy.example.com:5678
```

À configurer pour la session en cours uniquement

```
C:\> set HTTP_PROXY=http://username:password@proxy.example.com:1234
C:\> set HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Utilisation d'un proxy sur les EC2 instances Amazon

Si vous configurez un proxy sur une EC2 instance Amazon lancée avec un rôle IAM attaché, assurez-vous d'exempter l'adresse utilisée pour accéder aux métadonnées de l'instance. Pour ce faire, définissez la variable d'`NO_PROXY`environnement sur l'adresse IP du service de métadonnées d'instance, 169.254.169.254. Cette adresse ne varie pas.

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

À configurer pour toutes les sessions

```
C:\> setx NO_PROXY 169.254.169.254
```

À configurer pour la session en cours uniquement

```
C:\> set NO_PROXY=169.254.169.254
```

Résolution des problèmes

Si vous rencontrez des problèmes avec le AWS CLI, consultez les étapes the section called “Résoudre les erreurs” de résolution des problèmes. Pour connaître les étapes de résolution des problèmes les plus pertinentes, consultezthe section called “Erreurs de certificat SSL”.

Utilisation de points de terminaison dans le AWS CLI

Pour vous connecter par programmation à un Service AWS, vous utilisez un point de terminaison. Un point de terminaison est l'URL du point d'entrée d'un service AWS Web. Le AWS Command Line Interface (AWS CLI) utilise automatiquement le point de terminaison par défaut pour chaque service d'un Région AWS, mais vous pouvez spécifier un point de terminaison alternatif pour vos demandes d'API.

Sujets relatifs aux terminaux

- Définir le point de terminaison pour une seule commande
- Définir un point de terminaison global pour tous Services AWS
- Paramétré pour utiliser les FIPs points de terminaison pour tous Services AWS
- Conçu pour utiliser des points de terminaison à double pile pour tous Services AWS
- Définissez des points de terminaison spécifiques au service
  - Points de terminaison spécifiques au service : variables d'environnement
  - Points de terminaison spécifiques au service : fichier partagé config

  - Points de terminaison spécifiques au service : liste des identifiants spécifiques au service
- Points de terminaison basés sur des comptes
- Configuration du point de terminaison et priorité des paramètres

Définir le point de terminaison pour une seule commande

Pour remplacer les paramètres du point de terminaison ou les variables d'environnement pour une seule commande, utilisez l'option de ligne de `--endpoint-url` commande. L'exemple de commande suivant utilise une URL de point de terminaison Amazon S3 personnalisée.

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

Définir un point de terminaison global pour tous Services AWS

Pour acheminer les demandes relatives à tous les services vers une URL de point de terminaison personnalisée, utilisez l'un des paramètres suivants :

- Variables d'environnement :
  - AWS_IGNORE_CONFIGURED_ENDPOINT_URLS- Ignorez le point de terminaison configuré URLs.

    Linux or macOS

    ```
    $ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
    ```

    Windows Command Prompt

    À configurer pour toutes les sessions

    ```
    C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
    ```

    À configurer pour la session en cours uniquement

    ```
    C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
    ```

    PowerShell

    ```
    PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
    ```

- AWS_ENDPOINT_URL- Définissez l'URL globale du point de terminaison.

  Linux or macOS

  ```
  $ export AWS_ENDPOINT_URL=http://localhost:4567
  ```

  Windows Command Prompt

  À configurer pour toutes les sessions

  ```
  C:\> setx AWS_ENDPOINT_URL http://localhost:4567
  ```

  À configurer pour la session en cours uniquement

  ```
  C:\> set AWS_ENDPOINT_URL=http://localhost:4567
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_ENDPOINT_URL="http://localhost:4567"
  ```

- Le `config` dossier :
  - ignore_configure_endpoint_urls- Ignorez le point de terminaison configuré URLs.

    ```
    ignore_configure_endpoint_urls = true
    ```

  - endpoint_url- Définissez l'URL globale du point de terminaison.

    ```
    endpoint_url = http://localhost:4567
    ```

Les points de terminaison spécifiques au service et l'option de ligne de `--endpoint-url` commande remplacent tous les points de terminaison globaux.

Paramétré pour utiliser les FIPs points de terminaison pour tous Services AWS

Pour acheminer les demandes visant à ce que tous les services utilisent des FIPs points de terminaison, utilisez l'une des méthodes suivantes :

- AWS_USE_FIPS_ENDPOINTvariable d'environnement.

  Linux or macOS

  ```
  $ export AWS_USE_FIPS_ENDPOINT=true
  ```

  Windows Command Prompt

  À configurer pour toutes les sessions

  ```
  C:\> setx AWS_USE_FIPS_ENDPOINT true
  ```

  À configurer pour la session en cours uniquement

  ```
  C:\> set AWS_USE_FIPS_ENDPOINT=true
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_USE_FIPS_ENDPOINT="true"
  ```

- use_fips_endpointréglage du fichier.

  ```
  use_fips_endpoint = true
  ```

Certains AWS services proposent des points de terminaison compatibles avec la norme fédérale de traitement de l'information (FIPS) 140-2 dans certains cas. Régions AWS Lorsque le AWS service prend en charge le protocole FIPS, ce paramètre indique le point de terminaison FIPS à utiliser AWS CLI . Contrairement aux points de AWS terminaison standard, les points de terminaison FIPS utilisent une bibliothèque logicielle TLS conforme à la norme FIPS 140-2. Ces points de terminaison peuvent être requis par les entreprises qui interagissent avec le gouvernement des États-Unis.

Si ce paramètre est activé, mais qu'il n'existe pas de point de terminaison FIPS pour le service dans votre ordinateur Région AWS, la AWS commande risque d'échouer. Dans ce cas, spécifiez manuellement le point de terminaison à utiliser dans la commande à l'aide de l'--endpoint-urloption ou utilisez des points de terminaison spécifiques au service.

Pour plus d'informations sur la spécification des points de terminaison FIPS par Région AWS, consultez la section Points de terminaison FIPS par service.

Conçu pour utiliser des points de terminaison à double pile pour tous Services AWS

Pour acheminer les demandes visant à ce que tous les services utilisent des points de terminaison à double pile lorsqu'ils sont disponibles, utilisez l'un des paramètres suivants :

- AWS_USE_DUALSTACK_ENDPOINTvariable d'environnement.

  Linux or macOS

  ```
  $ export AWS_USE_DUALSTACK_ENDPOINT=true
  ```

  Windows Command Prompt

  À configurer pour toutes les sessions

  ```
  C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
  ```

  À configurer pour la session en cours uniquement

  ```
  C:\> set AWS_USE_DUALSTACK_ENDPOINT=true
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_USE_DUALSTACK_ENDPOINT="true"
  ```

- use_dualstack_endpointréglage du fichier.

  ```
  use_dualstack_endpoint = true
  ```

Permet d'utiliser des points de terminaison à double pile pour envoyer AWS des demandes. Pour en savoir plus sur les points de terminaison à double pile, qui prennent en charge à la fois le IPv6 trafic IPv4 et le trafic, consultez la section Utilisation des points de terminaison à double pile Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service. Des points de terminaison à double pile sont disponibles pour certains services dans certaines régions. S'il n'existe pas de point de terminaison à double pile pour le service ou Région AWS si la demande échoue. Par défaut, l'option est désactivée.

Définissez des points de terminaison spécifiques au service

La configuration du point de terminaison spécifique au service offre la possibilité d'utiliser un point de terminaison persistant de votre choix pour les AWS CLI demandes. Ces paramètres offrent la flexibilité nécessaire pour prendre en charge les points de terminaison locaux, les points de terminaison VPC et les environnements de développement locaux tiers AWS . Différents points de terminaison peuvent être utilisés pour les environnements de test et de production. Vous pouvez spécifier une URL de point de terminaison pour un individu Services AWS.

Les points de terminaison spécifiques au service peuvent être spécifiés de la manière suivante :

- L'option de ligne de commande `--endpoint-url` pour une seule commande.
- Variables d'environnement :
  - `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`- Ignorez tous les points de terminaison configurés URLs, sauf indication contraire sur la ligne de commande.
  - `AWS_ENDPOINT_URL_<SERVICE>`- Spécifie un point de terminaison personnalisé utilisé pour un service spécifique, où `<SERVICE>` il est remplacé par l' Service AWS identifiant. Pour toutes les variables spécifiques au service, voir. the section called “Liste des identifiants spécifiques au service”
- Fichier `config` :
  - `ignore_configure_endpoint_urls`- Ignorez tous les points de terminaison configurés URLs, sauf s'ils sont spécifiés à l'aide de variables d'environnement ou sur la ligne de commande.
  - `services`Section du `config` fichier associée au paramètre du `endpoint_url` fichier.

Rubriques relatives aux points de terminaison spécifiques aux services :

- Points de terminaison spécifiques au service : variables d'environnement
- Points de terminaison spécifiques au service : fichier partagé config
- Points de terminaison spécifiques au service : liste des identifiants spécifiques au service

Points de terminaison spécifiques au service : variables d'environnement

Les variables d'environnement remplacent les paramètres de votre fichier de configuration, mais ne remplacent pas les options spécifiées sur la ligne de commande. Utilisez des variables

d'environnement si vous souhaitez que tous les profils utilisent les mêmes points de terminaison sur votre appareil.

Les variables d'environnement spécifiques au service sont les suivantes :

- AWS_IGNORE_CONFIGURED_ENDPOINT_URLS- Ignorez tous les points de terminaison configurés URLs, sauf indication contraire sur la ligne de commande.

  Linux or macOS

  ```
  $ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
  ```

  Windows Command Prompt

  À configurer pour toutes les sessions

  ```
  C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
  ```

  À configurer pour la session en cours uniquement

  ```
  C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
  ```

- AWS_ENDPOINT_URL_<SERVICE>- Spécifie un point de terminaison personnalisé utilisé pour un service spécifique, où <SERVICE> il est remplacé par l' Service AWS identifiant. Pour toutes les variables spécifiques au service, voir. the section called “Liste des identifiants spécifiques au service”

  Les exemples de variables d'environnement suivants définissent un point de terminaison pour AWS Elastic Beanstalk :

  Linux or macOS

  ```
  $ export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
  ```

  Windows Command Prompt

  À configurer pour toutes les sessions

```
C:\> setx AWS_ENDPOINT_URL_ELASTIC_BEANSTALK http://localhost:4567
```

À configurer pour la session en cours uniquement

```
C:\> set AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL_ELASTIC_BEANSTALK="http://localhost:4567"
```

Pour plus d'informations sur la définition des variables d'environnement, consultezthe section called “Variables d'environnement”.

Points de terminaison spécifiques au service : fichier partagé `config`

Dans le `config` fichier partagé, `endpoint_url` est utilisé dans plusieurs sections. Pour définir un point de terminaison spécifique à un service, utilisez le `endpoint_url` paramètre imbriqué sous une clé d'identification de service dans une section. `services` Pour plus de détails sur la définition d'une `services` section dans votre `config` fichier partagé, consultezthe section called “`services`”.

L'exemple suivant utilise une `services` section pour configurer une URL de point de terminaison spécifique au service pour Amazon S3 et un point de terminaison global personnalisé utilisé pour tous les autres services :

```
[profile dev1]
endpoint_url = http://localhost:1234
services = s3-specific

[services testing-s3]
s3 =
  endpoint_url = http://localhost:4567
```

Un profil unique peut configurer des points de terminaison pour plusieurs services. L'exemple suivant définit le point de terminaison spécifique au service URLs pour Amazon S3 et AWS Elastic Beanstalk dans le même profil.

Pour obtenir la liste de toutes les clés d'identification de service à utiliser dans `services` cette section, consultezListe des identifiants spécifiques au service.

```
[profile dev1]
services = testing-s3-and-eb

[services testing-s3-and-eb]
s3 =
  endpoint_url = http://localhost:4567
elastic_beanstalk =
  endpoint_url = http://localhost:8000
```

La section de configuration du service peut être utilisée dans plusieurs profils. Dans l'exemple suivant, deux profils utilisent la même `services` définition :

```
[profile dev1]
output = json
services = testing-s3

[profile dev2]
output = text
services = testing-s3

[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567
```

Points de terminaison spécifiques au service : liste des identifiants spécifiques au service

L' Service AWS identifiant est basé sur le modèle d'API `serviceId` en remplaçant tous les espaces par des traits de soulignement et en minuscules toutes les lettres.

Le tableau suivant répertorie tous les identificateurs, clés de `config` fichier et variables d'environnement spécifiques au service.

Points de terminaison basés sur des comptes

Les points de terminaison basés sur un compte peuvent être spécifiés de la manière suivante :

- Variables d’environnement
  - `AWS_ACCOUNT_ID`- Spécifie l'ID du point de terminaison AWS basé sur le compte à utiliser pour les appels vers le service pris en charge. Services AWS

Linux or macOS

```
$ export AWS_ACCOUNT_ID=<account-id>
```

Windows Command Prompt

À configurer pour toutes les sessions

```
C:\> setx AWS_ACCOUNT_ID <account-id>
```

À configurer pour la session en cours uniquement

```
C:\> set AWS_ACCOUNT_ID=<account-id>
```

PowerShell

```
PS C:\> $Env:AWS_ACCOUNT_ID="<account-id>"
```

- AWS_ACCOUNT_ID_ENDPOINT_MODE- Spécifie s'il faut utiliser un point de terminaison AWS basé sur le compte IDs pour les appels vers le support. Services AWS Peut être réglé sur `preferred`, `disabled`, ou `required`. La valeur par défaut est `preferred`.

  Linux or macOS

  ```
  $ export AWS_ACCOUNT_ID_ENDPOINT_MODE=preferred
  ```

  Windows Command Prompt

  À configurer pour toutes les sessions

  ```
  C:\> setx AWS_ACCOUNT_ID_ENDPOINT_MODE preferred
  ```

  À configurer pour la session en cours uniquement

  ```
  C:\> set AWS_ACCOUNT_ID_ENDPOINT_MODE=preferred
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_ACCOUNT_ID_ENDPOINT_MODE="preferred"
  ```

- Le `config` dossier :
  - `aws_account_id`- Spécifie l'ID du point de terminaison AWS basé sur le compte à utiliser pour les appels vers le service pris en charge. Services AWS

    ```
    aws_account_id = <account-id>
    ```

  - `account_id_endpoint_mode`- Spécifie s'il faut utiliser un point de terminaison AWS basé sur le compte IDs pour les appels vers le support. Services AWS Peut être réglé sur `preferred`, `disabled` ou `required`. La valeur par défaut est préférée.

    ```
    account_id_endpoint_mode = preferred
    ```

Les terminaux basés sur des comptes contribuent à garantir des performances et une évolutivité élevées en utilisant votre identifiant de AWS compte pour rationaliser le routage des Service AWS demandes de services prenant en charge cette fonctionnalité. Lorsque vous utilisez un fournisseur d'informations d'identification et un service qui prend en charge les points de terminaison basés sur des comptes, ils construisent et utilisent AWS CLI automatiquement un point de terminaison basé sur un compte au lieu d'un point de terminaison régional.

Les points de terminaison basés sur un compte utilisent le format suivant, où `<account-id>` est remplacé par votre identifiant de AWS compte et `<region>` par votre : Région AWS

```
https://<account-id>.myservice.<region>.amazonaws.com
```

Par défaut AWS CLI, le mode de point de terminaison basé sur le compte est défini sur. `preferred`

Configuration du point de terminaison et priorité des paramètres

Les paramètres de configuration du point de terminaison se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. Les paramètres de configuration du point de AWS CLI terminaison sont prioritaires dans l'ordre suivant :

1. L'option `--endpoint-url` de ligne de commande.
2. Si cette option est activée, la variable d'environnement `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` global du point de terminaison ou le paramètre `ignore_configure_endpoint_urls` de profil ignore les points de terminaison personnalisés.

3. La valeur fournie par une variable d'environnement spécifique au serviceAWS_ENDPOINT_URL_<SERVICE>, telle que. AWS_ENDPOINT_URL_DYNAMODB
4. Les valeurs fournies par les variables d'AWS_ENDPOINT_URLenvironnement AWS_USE_DUALSTACK_ENDPOINTAWS_USE_FIPS_ENDPOINT, et.
5. La variable d'AWS_ACCOUNT_ID_ENDPOINT_MODEenvironnement est définie sur `preferred` ou à `required` l'aide de l'ID de compte dans la variable ou le aws_account_id paramètre d'AWS_ACCOUNT_IDenvironnement.
6. La valeur du point de terminaison spécifique au service fournie par le endpoint_url paramètre dans une `services` section du fichier partagé`config`.
7. La valeur fournie par le endpoint_url paramètre dans un `profile` `config` fichier partagé.
8. use_dualstack_endpointuse_fips_endpoint, et endpoint_url paramètres.
9. Le account_id_endpoint_mode paramètre est défini sur `preferred` ou à `required` l'aide de l'ID de compte dans la variable ou le aws_account_id paramètre d'AWS_ACCOUNT_IDenvironnement.
10. Toute URL de point de terminaison par défaut pour le point de terminaison correspondant Service AWS est utilisée en dernier. Pour afficher la liste des points de terminaison de service disponibles dans chaque région, consultez Régions et points de terminaison AWS dans le document Référence générale d'Amazon Web Services.

Authentification et informations d'accès pour AWS CLI

Vous devez établir la manière dont ils AWS CLI s'authentifient AWS lorsque vous développez avec des AWS services. Pour configurer les informations d'identification pour l'accès par programmation au AWS CLI, choisissez l'une des options suivantes. Les options sont présentées par ordre de recommandation.

Type d’authentification	Objectif	Instructions
Identifiants à court terme des utilisateurs du personnel d'IAM Identity Center	(Recommandé) Utilisez des informations d'identification à court terme pour un utilisateur du personnel d'IAM Identity Center. La meilleure pratique en matière de sécurité consiste AWS Organizations à utiliser IAM Identity Center. Il combine des informations d'identification à court terme avec un annuaire d'utilisateurs, tel que le répertoire intégré IAM Identity Center ou Active Directory.	the section called “Authentification IAM Identity Center”
Informations d'identification à court terme de l'utilisateur IAM	Utilisez les informations d'identification à court terme des utilisateurs IAM, qui sont plus sécurisées que les informations d'identification à long terme. Si vos informations d'identification sont compromises, elles peuvent être utilisées pendant	the section called “Identifiants à court terme”

Type d’authentification	Objectif	Instructions
	une durée limitée avant leur expiration.	
Utilisateurs d'IAM ou IAM Identity Center sur une instance Amazon EC2 .	Utilisez les métadonnées de l' EC2 instance Amazon pour demander des informations d'identification temporaires à l'aide du rôle attribué à l' EC2 instance Amazon.	the section called “ EC2 Métadonnées Amazon”
Assumer des rôles pour les autorisations	Associez une autre méthode d'identification et assumez un rôle pour un accès temporaire auquel Services AWS votre utilisateur n'a peut-être pas accès.	the section called “Rôles IAM”
Informations d'identification à long terme de l'utilisateur IAM	(Non recommandé) Utilisez des informations d'identification à long terme, qui n'ont pas d'expiration.	the section called “Utilisateurs IAM”
Stockage externe des utilisateurs du personnel d'IAM ou d'IAM Identity Center	(Non recommandé) Associez une autre méthode d'identification mais stockez les valeurs d'identification dans un emplacement extérieur à. AWS CLI La sécurité de cette méthode dépend de l'emplacement externe dans lequel les informations d'identification sont stockées.	the section called “External Credentials”

Configuration et priorité des informations d'identification

Les informations d'identification et les paramètres de configuration se trouvent à plusieurs endroits, tels que les variables d'environnement système ou utilisateur, les fichiers de AWS configuration locaux, ou sont explicitement déclarés sur la ligne de commande en tant que paramètre. Certaines authentifications ont priorité sur d'autres. Les paramètres AWS CLI d'authentification sont prioritaires dans l'ordre suivant :

1. Options de ligne de commande : remplace les paramètres de tout autre emplacement, tels que les `--profile` paramètres `--region``--output`, et.
2. Variables d'environnement : vous pouvez stocker des valeurs dans les variables d'environnement de votre système.
3. Assumer un rôle — Assumez les autorisations d'un rôle IAM par le biais de la configuration ou de la commande.
4. Assumer un rôle avec une identité Web : assumez les autorisations d'un rôle IAM en utilisant l'identité Web par le biais de la configuration ou de la commande.
5. AWS IAM Identity Center— Les paramètres de configuration d'IAM Identity Center stockés dans le `config` fichier sont mis à jour lorsque vous exécutez la `aws configure sso` commande. Les informations d'identification sont ensuite authentifiées lorsque vous exécutez la `aws sso login` commande. Le `config` fichier se trouve sous `~/.aws/config` Linux ou macOS, ou `C:\Users\`*`USERNAME`*`\.aws\config` sous Windows.
6. Fichier d'informations d'identification : le `config` fichier `credentials` et est mis à jour lorsque vous exécutez la commande`aws configure`. Le `credentials` fichier se trouve sous `~/.aws/credentials` Linux ou macOS, ou `C:\Users\`*`USERNAME`*`\.aws\credentials` sous Windows.
7. Processus personnalisé — Obtenez vos informations d'identification auprès d'une source externe.
8. Fichier de configuration — Le `config` fichier `credentials` et est mis à jour lorsque vous exécutez la commande`aws configure`. Le `config` fichier se trouve sous `~/.aws/config` Linux ou macOS, ou `C:\Users\`*`USERNAME`*`\.aws\config` sous Windows.
9. Informations d'identification du conteneur : vous pouvez associer un rôle IAM à chacune de vos définitions de tâches Amazon Elastic Container Service (Amazon ECS). Les informations d'identification temporaires pour ce rôle sont ensuite disponibles pour les conteneurs de cette tâche. Pour plus d'informations, consultez Rôles IAM pour les tâches dans le Guide du développeur Amazon Elastic Container Service.
10. Informations d'identification du profil d' EC2instance Amazon : vous pouvez associer un rôle IAM à chacune de vos instances Amazon Elastic Compute Cloud (Amazon EC2). Les informations

d'identification temporaires pour ce rôle sont ensuite disponibles pour l'exécution du code dans l'instance. Les informations d'identification sont fournies via le service de EC2 métadonnées Amazon. Pour plus d'informations, consultez les sections Rôles IAM pour Amazon EC2 dans le guide de l' EC2 utilisateur Amazon et Utilisation des profils d'instance dans le guide de l'utilisateur IAM.

Sujets supplémentaires dans cette section

- the section called “Authentification IAM Identity Center”
- the section called “Identifiants à court terme”
- the section called “Rôles IAM”
- the section called “Utilisateurs IAM”
- the section called “ EC2 Métadonnées Amazon”
- the section called “External Credentials”

Configuration de l'authentification IAM Identity Center à l'aide du AWS CLI

Cette rubrique fournit des instructions sur la façon de configurer le AWS CLI with AWS IAM Identity Center (IAM Identity Center) pour récupérer les informations d'identification permettant d'exécuter des AWS CLI commandes. Il existe principalement deux méthodes pour authentifier les utilisateurs auprès d'IAM Identity Center afin d'obtenir les informations d'identification nécessaires pour exécuter des AWS CLI commandes via le `config` fichier :

- Configuration du fournisseur de jetons SSO (recommandée).
- Configuration ancienne non actualisable.

Pour plus d'informations sur l'utilisation de l'authentification au porteur, qui n'utilise aucun identifiant de compte ni rôle, consultez la section Configuration pour utiliser l'authentification AWS CLI avec CodeCatalyst dans le guide de CodeCatalyst l'utilisateur Amazon.

> **Note**
>
> Pour un processus guidé d'utilisation d'IAM Identity Center avec des AWS CLI commandes, voirthe section called “Tutoriel : AWS IAM Identity Center et Amazon S3”.

Rubriques

- the section called “Prérequis”
- the section called “Configurez votre profil avec l'`aws configure sso`assistant”
- the section called “Configurez uniquement votre `sso-session` section avec `aws configure sso-session` l'assistant”
- the section called “Configuration manuelle à l'aide du `config` fichier”
- the section called “Connectez-vous à une session IAM Identity Center”
- the section called “Exécutez une commande avec votre profil IAM Identity Center”
- the section called “Déconnectez-vous de vos sessions IAM Identity Center”
- the section called “Résolution des problèmes”
- the section called “Ressources connexes”

Prérequis

- Installez le AWS CLI. Pour de plus amples informations, veuillez consulter the section called “Installation/mise à jour”.
- Vous devez d'abord avoir accès à l'authentification SSO dans IAM Identity Center. Choisissez l'une des méthodes suivantes pour accéder à vos AWS informations d'identification.

Je n'ai pas établi d'accès via IAM Identity Center

Suivez les instructions de la section Mise en route du guide de AWS IAM Identity Center l'utilisateur. Ce processus active IAM Identity Center, crée un utilisateur administratif et ajoute un ensemble d'autorisations de moindre privilège approprié.

> **Note**
>
> Créez un ensemble d'autorisations qui applique les autorisations du moindre privilège. Nous vous recommandons d'utiliser l'ensemble `PowerUserAccess` d'autorisations prédéfini, sauf si votre employeur a créé un ensemble d'autorisations personnalisé à cette fin.

Quittez le portail et reconnectez-vous pour voir vos Comptes AWS informations d'accès programmatiques et les options pour `Administrator` ou`PowerUserAccess`. Sélectionnez `PowerUserAccess` lorsque vous travaillez avec le SDK.

J'y ai déjà accès AWS via un fournisseur d'identité fédéré géré par mon employeur (tel qu'Azure AD ou Okta)

Connectez-vous AWS via le portail de votre fournisseur d'identité. Si votre administrateur cloud vous a accordé des autorisations `PowerUserAccess` (de développeur), vous voyez Comptes AWS celles auxquelles vous avez accès et votre ensemble d'autorisations. À côté du nom de votre ensemble d'autorisations, vous trouverez des options permettant d'accéder aux comptes manuellement ou par programmation à l'aide de cet ensemble d'autorisations.

Les implémentations personnalisées peuvent entraîner des expériences différentes, telles que des noms d'ensembles d'autorisations différents. Si vous ne savez pas quel ensemble d'autorisations utiliser, contactez votre équipe informatique pour obtenir de l'aide.

J'y ai déjà accès AWS via le portail AWS d'accès géré par mon employeur

Connectez-vous AWS via votre portail AWS d'accès. Si votre administrateur cloud vous a accordé des autorisations `PowerUserAccess` (de développeur), vous voyez Comptes AWS celles auxquelles vous avez accès et votre ensemble d'autorisations. À côté du nom de votre ensemble d'autorisations, vous trouverez des options permettant d'accéder aux comptes manuellement ou par programmation à l'aide de cet ensemble d'autorisations.

J'y ai déjà accès AWS via un fournisseur d'identité personnalisé fédéré géré par mon employeur

Contactez votre équipe informatique pour obtenir de l'aide.

Après avoir accédé à IAM Identity Center, collectez les informations de votre IAM Identity Center en effectuant les opérations suivantes :

1. Rassemblez vos `SSO Region` valeurs `SSO Start URL` et celles dont vous avez besoin pour fonctionner `aws configure sso`

a. Dans votre portail AWS d'accès, sélectionnez l'ensemble d'autorisations que vous utilisez pour le développement, puis cliquez sur le lien Clés d'accès.

b. Dans la boîte de dialogue Obtenir les informations d'identification, choisissez l'onglet correspondant à votre système d'exploitation.

c. Choisissez la méthode d'identification IAM Identity Center pour obtenir les `SSO Region` valeurs `SSO Start URL` et.

2. À partir de la version 2.22.0, vous pouvez également utiliser l'URL de l'émetteur au lieu de l'URL de démarrage. L'URL de l'émetteur se trouve dans la AWS IAM Identity Center console à l'un des emplacements suivants :

   - Sur la page Tableau de bord, l'URL de l'émetteur se trouve dans le résumé des paramètres.
   - Sur la page Paramètres, l'URL de l'émetteur se trouve dans les paramètres de la source d'identité.

3. Pour plus d'informations sur la valeur d'étendue à enregistrer, consultez la section Étendue d'accès OAuth 2.0 dans le guide de l'utilisateur d'IAM Identity Center.

Configurez votre profil avec l'**`aws configure sso`**assistant

Pour configurer un profil IAM Identity Center pour votre AWS CLI :

1. Dans le terminal de votre choix, exécutez la `aws configure sso` commande.

   (Recommended) IAM Identity Center

   Créez un nom de session, indiquez l'URL de démarrage de votre IAM Identity Center ou l'URL de l'émetteur, Région AWS celle qui héberge le répertoire du IAM Identity Center et le périmètre d'enregistrement.

   ```
   $ aws configure sso
   SSO session name (Recommended): my-sso
   SSO start URL [None]: https://my-sso-portal.awsapps.com/start
   SSO region [None]: us-east-1
   SSO registration scopes [None]: sso:account:access
   ```

   L'autorisation PKCE (Proof Key for Code Exchange) est utilisée par défaut à AWS CLI partir de la version 2.22.0 et doit être utilisée sur les appareils dotés d'un navigateur. Pour continuer à utiliser l'autorisation de l'appareil, ajoutez l'`--use-device-code`option.

```
$ aws configure sso --use-device-code
```

Legacy IAM Identity Center

Ignorez le nom de session et indiquez l'URL de démarrage de votre centre d'identité IAM et la AWS région qui héberge le répertoire du centre d'identité.

```
$ aws configure sso
SSO session name (Recommended):
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1
```

2. Les AWS CLI tentatives d'ouverture de votre navigateur par défaut pour le processus de connexion de votre compte IAM Identity Center. Ce processus peut vous demander d'autoriser l' AWS CLI accès à vos données. Étant donné que le AWS CLI est construit au-dessus du SDK pour Python, les messages d'autorisation peuvent contenir des variantes du `botocore` nom.

   - S'il n'est AWS CLI pas possible d'ouvrir le navigateur, les instructions pour démarrer manuellement le processus de connexion s'affichent en fonction du type d'autorisation que vous utilisez.

     PKCE authorization

     L'autorisation Proof Key for Code Exchange (PKCE) est utilisée par défaut à AWS CLI partir de la version 2.22.0. L'URL affichée est une URL unique commençant par : *https://oidc.us-east-1.amazonaws.com/authorize*

     L'autorisation PKCE URLs doit être ouverte sur le même appareil que celui sur lequel vous vous connectez et doit être utilisée pour un appareil doté d'un navigateur.

```
Attempting to automatically open the SSO authorization
 page in your
default browser.
If the browser does not open or you wish to use a
 different device to
authorize the request, open the following URL:

https://oidc.us-east-1.amazonaws.com/authorize?
<abbreviated>
```

Device authorization

L'autorisation de l'appareil OAuth 2.0 est utilisée par AWS CLI les versions antérieures à 2.22.0. Vous pouvez activer cette méthode sur les versions plus récentes à l'aide de l'`--use-device-code`option.

L'autorisation de l'appareil URLs n'a pas besoin d'être ouverte sur le même appareil que celui sur lequel vous vous connectez et peut être utilisée pour un appareil avec ou sans navigateur.

```
If the browser does not open or you wish to use a
 different device to
authorize this request, open the following URL:
https://device.sso.us-west-2.amazonaws.com/

Then enter the code:
QCFK-N451
```

3. Sélectionnez le AWS compte à utiliser dans la liste qui s'affiche. Si vous n'êtes autorisé à utiliser qu'un seul compte, celui-ci est AWS CLI automatiquement sélectionné et ignore l'invite.

```
There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (123456789011)
  ProductionAccount, production-account-admin@example.com (123456789022)
```

4. Sélectionnez le rôle IAM à utiliser dans la liste affichée. S'il n'y a qu'un seul rôle disponible, il le sélectionne AWS CLI automatiquement et ignore l'invite.

```
Using the account ID 123456789011
There are 2 roles available to you.
> ReadOnly
  FullAccess
```

5. Spécifiez le format de sortie par défaut Région AWS, le format par défaut auquel envoyer les commandes et le nom du profil. Si vous le spécifiez `default` comme nom de profil, ce profil devient le profil par défaut utilisé. Dans l'exemple ci-dessous, l'utilisateur entre une région par défaut, un format de sortie par défaut et le nom du profil.

```
Default client Region [None]: us-west-2<ENTER>
CLI default output format (json if not specified) [None]: json<ENTER>
```

```
Profile name [123456789011_ReadOnly]: my-dev-profile<ENTER>
```

6. Un message final décrit la configuration de profil terminée. Vous pouvez désormais utiliser ce profil pour demander des informations d'identification. Utilisez la `aws sso login` commande pour demander et récupérer les informations d'identification nécessaires pour exécuter des commandes. Pour obtenir des instructions, veuillez consulter Connectez-vous à une session IAM Identity Center.

Fichier de configuration généré

Ces étapes aboutissent à la création de la `sso-session` section et du profil nommé dans le `config` fichier qui se présentent comme suit :

IAM Identity Center

```
[profile my-dev-profile]
sso_session = my-sso
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Legacy IAM Identity Center

```
[profile my-dev-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
output = json
```

Configurez uniquement votre **`sso-session`** section avec **`aws configure sso-session`** l'assistant

> **Note**
>
> Cette configuration n'est pas compatible avec l'ancien IAM Identity Center.

La `aws configure sso-session` commande met à jour les `sso-session` sections du `~/.aws/config` fichier. Exécutez la `aws configure sso-session` commande et indiquez l'URL de démarrage ou l'URL de l'émetteur de votre IAM Identity Center ainsi que la AWS région qui héberge le répertoire de l'IAM Identity Center.

```
$ aws configure sso-session
SSO session name: my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

Configuration manuelle à l'aide du **`config`** fichier

Les informations de configuration d'IAM Identity Center sont stockées dans le `config` fichier et peuvent être modifiées à l'aide d'un éditeur de texte. Pour ajouter manuellement le support IAM Identity Center à un profil nommé, vous devez ajouter des clés et des valeurs au `config` fichier.

Fichier de configuration d'IAM Identity Center

La `sso-session` section du `config` fichier est utilisée pour regrouper les variables de configuration permettant d'acquérir des jetons d'accès SSO, qui peuvent ensuite être utilisés pour acquérir des AWS informations d'identification. Les paramètres suivants sont utilisés :

- (Obligatoire) `sso_start_url`
- (Obligatoire) `sso_region`
- `sso_account_id`
- `sso_role_name`
- `sso_registration_scopes`

Vous définissez une `sso-session` section et vous l'associez à un profil. Les `sso_start_url` paramètres `sso_region` et doivent être définis dans la `sso-session` section. Généralement, `sso_account_id` et `sso_role_name` doit être défini dans la `profile` section afin que le SDK puisse demander des informations d'identification SSO.

L'exemple suivant configure le SDK pour demander des informations d'identification SSO et prend en charge l'actualisation automatique des jetons :

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

Cela permet également de réutiliser les `sso-session` configurations sur plusieurs profils :

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

Cependant, ils `sso_account_id` `sso_role_name` ne sont pas obligatoires pour tous les scénarios de configuration de jetons SSO. Si votre application utilise uniquement des AWS services qui prennent en charge l'authentification par porteur, les informations d' AWS identification traditionnelles ne sont pas nécessaires. L'authentification du porteur est un schéma d'authentification HTTP qui utilise des jetons de sécurité appelés jetons porteurs. Dans ce scénario, `sso_account_id` et `sso_role_name` ne sont pas obligatoires. Consultez le guide individuel de votre AWS service pour déterminer s'il prend en charge l'autorisation par jeton au porteur.

De plus, les étendues d'enregistrement peuvent être configurées dans le cadre d'un`sso-session`. Scope est un mécanisme de la OAuth version 2.0 qui limite l'accès d'une application au compte d'un utilisateur. Une application peut demander une ou plusieurs étendues, et le jeton d'accès délivré à l'application sera limité aux étendues accordées. Ces étendues définissent les autorisations demandées pour être autorisées pour le client OIDC enregistré et les jetons d'accès récupérés par le client. L'exemple suivant fournit un accès `sso_registration_scopes` pour répertorier les comptes/rôles :

```
[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Le jeton d'authentification est mis en cache sur le disque sous le `sso/cache` répertoire avec un nom de fichier basé sur le nom de session.

Fichier de configuration de l'ancien IAM Identity Center

> **Note**
>
> L'actualisation automatique des jetons n'est pas prise en charge avec l'ancienne configuration non actualisable. Nous vous recommandons d'utiliser la configuration du jeton SSO.

Pour ajouter manuellement le support IAM Identity Center à un profil nommé, vous devez ajouter les clés et valeurs suivantes à la définition du profil dans le `config` fichier.

- `sso_start_url`
- `sso_region`
- `sso_account_id`
- `sso_role_name`

Vous pouvez inclure toutes les autres clés et valeurs valides dans le `.aws/config` fichier. L'exemple suivant est un profil IAM Identity Center :

```
[profile my-sso-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
```

```
sso_region = us-west-2
sso_account_id = 111122223333
sso_role_name = SSOReadOnlyRole
region = us-west-2
output = json
```

Pour exécuter des commandes, vous devez the section called “Connectez-vous à une session IAM Identity Center” d'abord demander et récupérer vos informations d'identification temporaires.

Pour plus d'informations sur les `credentials` fichiers `config` et, consultezthe section called “Paramètres de configuration”.

Connectez-vous à une session IAM Identity Center

> **Note**
>
> Le processus de connexion peut vous demander d'autoriser l' AWS CLI accès à vos données. Étant donné que le AWS CLI est construit au-dessus du SDK pour Python, les messages d'autorisation peuvent contenir des variantes du `botocore` nom.

Pour récupérer et mettre en cache un ensemble d'informations d'identification IAM Identity Center, exécutez la commande suivante pour AWS CLI ouvrir votre navigateur par défaut et vérifier votre connexion à IAM Identity Center.

```
$ aws sso login --profile my-dev-profile
SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.
Successfully logged into Start URL: https://my-sso-portal.awsapps.com/start
```

Les informations d'identification de votre session IAM Identity Center sont mises en cache et les AWS CLI utilisent pour récupérer en toute sécurité les AWS informations d'identification pour le rôle IAM spécifié dans le profil.

Si vous ne AWS CLI parvenez pas à ouvrir votre navigateur

S'il n'est AWS CLI pas possible d'ouvrir automatiquement votre navigateur, les instructions pour démarrer manuellement le processus de connexion s'affichent en fonction du type d'autorisation que vous utilisez.

PKCE authorization

L'autorisation Proof Key for Code Exchange (PKCE) est utilisée par défaut à AWS CLI partir de la version 2.22.0. L'URL affichée est une URL unique commençant par : *https://oidc.us-east-1.amazonaws.com/authorize*

L'autorisation PKCE URLs doit être ouverte sur le même appareil que celui sur lequel vous vous connectez et doit être utilisée pour un appareil doté d'un navigateur.

```
Attempting to automatically open the SSO authorization
 page in your
default browser.
If the browser does not open or you wish to use a
 different device to
authorize the request, open the following URL:

https://oidc.us-east-1.amazonaws.com/authorize?
<abbreviated>
```

Device authorization

L'autorisation de l'appareil OAuth 2.0 est utilisée par AWS CLI les versions antérieures à 2.22.0. Vous pouvez activer cette méthode sur les versions plus récentes à l'aide de l'`--use-device-code`option.

L'autorisation de l'appareil URLs n'a pas besoin d'être ouverte sur le même appareil que celui sur lequel vous vous connectez et peut être utilisée pour un appareil avec ou sans navigateur.

```
If the browser does not open or you wish to use a
 different device to
authorize this request, open the following URL:
https://device.sso.us-west-2.amazonaws.com/

Then enter the code:
QCFK-N451
```

Vous pouvez également spécifier le `sso-session` profil à utiliser lors de la connexion à l'aide du `--sso-session` paramètre de la `aws sso login` commande. L'`sso-session`option n'est pas disponible pour l'ancien IAM Identity Center.

```
$ aws sso login --sso-session my-dev-session
```

À partir de la version 2.22.0, l'autorisation PKCE est la valeur par défaut. Pour utiliser l'autorisation de l'appareil pour vous connecter, ajoutez l'`--use-device-code`option.

```
$ aws sso login --profile my-dev-profile --use-device-code
```

Le jeton d'authentification est mis en cache sur le disque sous le `~/.aws/sso/cache` répertoire avec un nom de fichier basé sur le`sso_start_url`.

Exécutez une commande avec votre profil IAM Identity Center

Une fois connecté, vous pouvez utiliser vos informations d'identification pour appeler des AWS CLI commandes avec le profil nommé associé. L'exemple suivant montre une commande utilisant un profil :

```
$ aws sts get-caller-identity --profile my-dev-profile
```

Tant que vous êtes connecté à IAM Identity Center et que ces informations d'identification mises en cache n'ont pas expiré, les informations d'identification expirées sont AWS CLI automatiquement renouvelées en cas de besoin AWS . Toutefois, si vos informations d'identification IAM Identity Center expirent, vous devez les renouveler explicitement en vous reconnectant à votre compte IAM Identity Center.

Déconnectez-vous de vos sessions IAM Identity Center

Lorsque vous avez terminé d'utiliser votre profil IAM Identity Center, vous pouvez laisser vos informations d'identification expirer ou exécuter la commande suivante pour supprimer vos informations d'identification mises en cache.

```
$ aws sso logout
Successfully signed out of all SSO profiles.
```

Résolution des problèmes

Si vous rencontrez des problèmes lors de l'utilisation du AWS CLI, consultez les étapes the section called “Résoudre les erreurs” de résolution des problèmes.

Ressources connexes

Les ressources supplémentaires sont les suivantes.

- the section called “Concepts du centre d'identité IAM”
- the section called “Tutoriel : AWS IAM Identity Center et Amazon S3”
- the section called “Installation/mise à jour”
- the section called “Paramètres de configuration”
- `aws configure sso`dans la AWS CLI version 2 Reference
- `aws configure sso-session`dans la AWS CLI version 2 Reference
- `aws sso login`dans la AWS CLI version 2 Reference
- `aws sso logout`dans la AWS CLI version 2 Reference
- Configuration pour utiliser le AWS CLI with CodeCatalyst dans le guide de CodeCatalyst l'utilisateur Amazon
- OAuth Étendue d'accès 2.0 dans le guide de l'utilisateur d'IAM Identity Center
- Tutoriels de mise en route dans le guide de l'utilisateur d'IAM Identity Center

AWS IAM Identity Center concepts pour le AWS CLI

Cette rubrique décrit les concepts clés de AWS IAM Identity Center (IAM Identity Center). IAM Identity Center est un service IAM basé sur le cloud qui simplifie la gestion de l'accès des utilisateurs à de multiples Comptes AWS applications et outils en s'intégrant aux fournisseurs d'identité (IdP) existants. SDKs Il permet l'authentification unique sécurisée, la gestion des autorisations et l'audit via un portail utilisateur centralisé, rationalisant ainsi la gouvernance des identités et des accès pour les entreprises.

Rubriques

- Qu'est-ce que IAM Identity Center
- Conditions
- Comment fonctionne IAM Identity Center
- Ressources supplémentaires

Qu'est-ce que IAM Identity Center

IAM Identity Center est un service de gestion des identités et des accès (IAM) basé sur le cloud qui vous permet de gérer de manière centralisée l'accès à de multiples Comptes AWS applications professionnelles.

Il fournit un portail utilisateur où les utilisateurs autorisés peuvent accéder aux applications Comptes AWS et aux applications pour lesquelles ils ont été autorisés, en utilisant leurs informations d'identification professionnelles existantes. Cela permet aux entreprises d'appliquer des politiques de sécurité cohérentes et de rationaliser la gestion des accès des utilisateurs.

Quel que soit l'IdP que vous utilisez, IAM Identity Center élimine ces distinctions. Par exemple, vous pouvez connecter Microsoft Azure AD comme décrit dans l'article de blog The Next Evolution in IAM Identity Center.

> **Note**
>
> Pour plus d'informations sur l'utilisation de l'authentification au porteur, qui n'utilise aucun identifiant de compte ni rôle, consultez la section Configuration pour utiliser l'authentification AWS CLI avec CodeCatalyst dans le guide de CodeCatalyst l'utilisateur Amazon.

Conditions

Les termes courants utilisés lors de l'utilisation d'IAM Identity Center sont les suivants :

Fournisseur d'identité

: Un système de gestion des identités tel que IAM Identity Center, Microsoft Azure AD, Okta ou votre propre service d'annuaire d'entreprise.

AWS IAM Identity Center

: IAM Identity Center est le service IdP AWS détenu. Anciennement connu sous le AWS nom de Single Sign-On, SDKs les outils conservent les espaces de noms des `sso` API pour des raisons de rétrocompatibilité. Pour plus d'informations, voir Renommer le centre d'identité IAM dans le guide de l'AWS IAM Identity Center utilisateur.

Portail d'accès AWS URL, URL de démarrage SSO, URL de démarrage

: URL unique du centre d'identité IAM de votre organisation pour accéder à vos services Comptes AWS, ressources et services autorisés.

URL de l'émetteur

URL unique de l'émetteur de l'IAM Identity Center de votre organisation pour un accès programmatique à vos services Comptes AWS, ressources et autorisations. À partir de la version 2.22.0 de AWS CLI, l'URL de l'émetteur peut être utilisée de manière interchangeable avec l'URL de départ.

Fédération

Processus d'établissement de la confiance entre IAM Identity Center et un fournisseur d'identité afin de permettre l'authentification unique (SSO).

Comptes AWS

Le via Comptes AWS lequel vous fournissez l'accès aux utilisateurs AWS IAM Identity Center.

Ensembles d'autorisations, AWS informations d'identification, informations d'identification, informations d'identification sigv4

Collections prédéfinies d'autorisations qui peuvent être attribuées à des utilisateurs ou à des groupes pour qu'ils accordent l'accès Services AWS.

Champs d'enregistrement, champs d'accès, champs d'application

Les scopes sont un mécanisme de la OAuth version 2.0 qui limite l'accès d'une application au compte d'un utilisateur. Une application peut demander une ou plusieurs étendues, et le jeton d'accès délivré à l'application est limité aux étendues accordées. Pour plus d'informations sur les étendues, consultez la section Étendue d'accès dans le guide de l'utilisateur d'IAM Identity Center.

Jetons, jeton d'actualisation, jeton d'accès

Les jetons sont des identifiants de sécurité temporaires qui vous sont délivrés lors de l'authentification. Ces jetons contiennent des informations sur votre identité et les autorisations qui vous ont été accordées.

Lorsque vous accédez à une AWS ressource ou à une application via le portail IAM Identity Center, votre jeton est présenté à des AWS fins d'authentification et d'autorisation. Cela permet AWS de vérifier votre identité et de vous assurer que vous disposez des autorisations nécessaires pour effectuer les actions demandées.

Le jeton d'authentification est mis en cache sur le disque sous le `~/.aws/sso/cache` répertoire avec un nom de fichier JSON basé sur le nom de session.

Session

Une session IAM Identity Center fait référence à la période pendant laquelle un utilisateur est authentifié et autorisé à accéder à des AWS ressources ou à des applications. Lorsqu'un utilisateur se connecte au portail IAM Identity Center, une session est établie et le jeton de l'utilisateur est valide pendant une durée spécifiée. Pour plus d'informations sur la définition de la durée des sessions, voir Définir la durée des sessions dans le Guide de l'AWS IAM Identity Center utilisateur.

Au cours de la session, vous pouvez naviguer entre différents AWS comptes et applications sans avoir à vous authentifier à nouveau, tant que leur session reste active. Lorsque la session expire, reconnectez-vous pour renouveler votre accès.

Les sessions IAM Identity Center contribuent à offrir une expérience utilisateur fluide tout en appliquant les meilleures pratiques de sécurité en limitant la validité des informations d'accès des utilisateurs.

Octroi de code d'autorisation avec PKCE, PKCE, clé de preuve pour l'échange de code

À partir de la version 2.22.0, Proof Key for Code Exchange (PKCE) est un flux d'autorisation d'authentification OAuth 2.0 pour les appareils dotés d'un navigateur. PKCE est un moyen simple et sûr de vous authentifier et d'obtenir le consentement pour accéder à vos AWS ressources à partir d'ordinateurs de bureau et d'appareils mobiles équipés de navigateurs Web. Il s'agit du comportement d'autorisation par défaut. Pour plus d'informations sur le PKCE, consultez la section Octroi de code d'autorisation avec PKCE dans le guide de l'AWS IAM Identity Center utilisateur.

Octroi d'autorisation d'appareil

Un flux d'autorisation d'authentification OAuth 2.0 pour les appareils avec ou sans navigateur Web. Pour plus d'informations sur la définition de la durée des sessions, consultez la section Octroi d'autorisation d'appareil dans le guide de AWS IAM Identity Center l'utilisateur.

Comment fonctionne IAM Identity Center

IAM Identity Center s'intègre au fournisseur d'identité de votre organisation, tel que IAM Identity Center, Microsoft Azure AD ou Okta. Les utilisateurs s'authentifient auprès de ce fournisseur d'identité, et IAM Identity Center associe ensuite ces identités aux autorisations et aux accès appropriés au sein de votre AWS environnement.

Le flux de travail IAM Identity Center suivant suppose que vous avez déjà configuré votre système AWS CLI pour utiliser IAM Identity Center :

1. Dans le terminal de votre choix, exécutez la `aws sso login` commande.
2. Connectez-vous à votre compte Portail d'accès AWS pour démarrer une nouvelle session.
   - Lorsque vous démarrez une nouvelle session, vous recevez un jeton d'actualisation et un jeton d'accès mis en cache.
   - Si vous avez déjà une session active, la session existante est réutilisée et expire en même temps que la session existante.
3. Sur la base du profil que vous avez défini dans votre `config` fichier, IAM Identity Center utilise les ensembles d'autorisations appropriés, accordant l'accès aux applications Comptes AWS et aux applications pertinentes.
4. Les outils AWS CLI SDKs, et utilisent votre rôle IAM supposé pour effectuer des appels, par Services AWS exemple pour créer des compartiments Amazon S3 jusqu'à l'expiration de cette session.
5. Le jeton d'accès d'IAM Identity Center est vérifié toutes les heures et est automatiquement actualisé à l'aide du jeton d'actualisation.
   - Si le jeton d'accès est expiré, le SDK ou l'outil utilise le jeton d'actualisation pour obtenir un nouveau jeton d'accès. Les durées de session de ces jetons sont ensuite comparées, et si le jeton d'actualisation n'est pas expiré, IAM Identity Center fournit un nouveau jeton d'accès.
   - Si le jeton d'actualisation a expiré, aucun nouveau jeton d'accès n'est fourni et votre session est terminée.
6. Les sessions se terminent après l'expiration des jetons d'actualisation ou lorsque vous vous déconnectez manuellement à l'aide de la `aws sso logout` commande. Les informations d'identification mises en cache sont supprimées. Pour continuer à accéder aux services via IAM Identity Center, vous devez démarrer une nouvelle session à l'aide de la `aws sso login` commande.

Ressources supplémentaires

Les ressources supplémentaires sont les suivantes.

- the section called “Authentification IAM Identity Center”
- the section called “Tutoriel : AWS IAM Identity Center et Amazon S3”
- the section called “Installation/mise à jour”

- the section called “Paramètres de configuration”
- `aws configure sso`dans la AWS CLI version 2 Reference
- `aws configure sso-session`dans la AWS CLI version 2 Reference
- `aws sso login`dans la AWS CLI version 2 Reference
- `aws sso logout`dans la AWS CLI version 2 Reference
- Configuration pour utiliser le AWS CLI with CodeCatalyst dans le guide de CodeCatalyst l'utilisateur Amazon
- Renommer le centre d'identité IAM dans le guide de l'utilisateur AWS IAM Identity Center
- OAuth Étendue d'accès 2.0 dans le guide de l'utilisateur d'IAM Identity Center
- Définissez la durée de la session dans le guide de AWS IAM Identity Center l'utilisateur
- Tutoriels de mise en route dans le guide de l'utilisateur d'IAM Identity Center

Tutoriel : Utilisation d'IAM Identity Center pour exécuter des commandes Amazon S3 dans le AWS CLI

Cette rubrique décrit comment configurer les commandes AWS CLI pour authentifier les utilisateurs auprès de Current AWS IAM Identity Center (IAM Identity Center) afin de récupérer les informations d'identification pour exécuter AWS Command Line Interface (AWS CLI) les commandes Amazon Simple Storage Service (Amazon S3).

Rubriques

- Étape 1 : Authentification dans le centre d'identité IAM
- Étape 2 : Rassemblez les informations de votre centre d'identité IAM
- Étape 3 : créer des compartiments Amazon S3
- Étape 4 : installez le AWS CLI
- Étape 5 : Configuration de votre AWS CLI profil
- Étape 6 : Connectez-vous à IAM Identity Center
- Étape 7 : Exécuter les commandes Amazon S3
- Étape 8 : Déconnectez-vous d'IAM Identity Center
- Étape 9 : Nettoyer les ressources
- Résolution des problèmes
- Ressources supplémentaires

Étape 1 : Authentification dans le centre d'identité IAM

Accédez à l'authentification SSO dans IAM Identity Center. Choisissez l'une des méthodes suivantes pour accéder à vos AWS informations d'identification.

Je n'ai pas établi d'accès via IAM Identity Center

Suivez les instructions de la section Mise en route du guide de AWS IAM Identity Center l'utilisateur. Ce processus active IAM Identity Center, crée un utilisateur administratif et ajoute un ensemble d'autorisations de moindre privilège approprié.

> **Note**
>
> Créez un ensemble d'autorisations qui applique les autorisations du moindre privilège. Nous vous recommandons d'utiliser l'ensemble `PowerUserAccess` d'autorisations prédéfini, sauf si votre employeur a créé un ensemble d'autorisations personnalisé à cette fin.

Quittez le portail et reconnectez-vous pour voir vos Comptes AWS informations d'accès programmatiques et les options pour `Administrator` ou`PowerUserAccess`. Sélectionnez `PowerUserAccess` lorsque vous travaillez avec le SDK.

J'y ai déjà accès AWS via un fournisseur d'identité fédéré géré par mon employeur (tel qu'Azure AD ou Okta)

Connectez-vous AWS via le portail de votre fournisseur d'identité. Si votre administrateur cloud vous a accordé des autorisations `PowerUserAccess` (de développeur), vous voyez Comptes AWS celles auxquelles vous avez accès et votre ensemble d'autorisations. À côté du nom de votre ensemble d'autorisations, vous pouvez accéder aux comptes manuellement ou par programmation à l'aide de cet ensemble d'autorisations.

Les implémentations personnalisées peuvent entraîner des expériences différentes, telles que des noms d'ensembles d'autorisations différents. Si vous ne savez pas quel ensemble d'autorisations utiliser, contactez votre équipe informatique pour obtenir de l'aide.

J'y ai déjà accès AWS via le portail AWS d'accès géré par mon employeur

Connectez-vous AWS via votre portail AWS d'accès. Si votre administrateur cloud vous a accordé des autorisations `PowerUserAccess` (de développeur), vous voyez Comptes AWS celles auxquelles

vous avez accès et votre ensemble d'autorisations. À côté du nom de votre ensemble d'autorisations, vous pouvez accéder aux comptes manuellement ou par programmation à l'aide de cet ensemble d'autorisations.

J'y ai déjà accès AWS via un fournisseur d'identité personnalisé fédéré géré par mon employeur

Contactez votre équipe informatique pour obtenir de l'aide.

Étape 2 : Rassemblez les informations de votre centre d'identité IAM

Après avoir obtenu l'accès à AWS, collectez les informations de votre centre d'identité IAM en effectuant les opérations suivantes :

1. Rassemblez vos `SSO Region` valeurs `SSO Start URL` et celles dont vous avez besoin pour fonctionner `aws configure sso`

    a. Dans votre portail AWS d'accès, sélectionnez l'ensemble d'autorisations que vous utilisez pour le développement, puis cliquez sur le lien Clés d'accès.

    b. Dans la boîte de dialogue Obtenir les informations d'identification, choisissez l'onglet correspondant à votre système d'exploitation.

    c. Choisissez la méthode d'identification IAM Identity Center pour obtenir les `SSO Region` valeurs `SSO Start URL` et.
2. À partir de la version 2.22.0, vous pouvez également utiliser la nouvelle URL de l'émetteur au lieu de l'URL de départ. L'URL de l'émetteur se trouve dans la AWS IAM Identity Center console à l'un des emplacements suivants :
    - Sur la page Tableau de bord, l'URL de l'émetteur se trouve dans le résumé des paramètres.
    - Sur la page Paramètres, l'URL de l'émetteur se trouve dans les paramètres de la source d'identité.
3. Pour plus d'informations sur la valeur d'étendue à enregistrer, consultez la section Étendue d'accès OAuth 2.0 dans le guide de l'utilisateur d'IAM Identity Center.

Étape 3 : créer des compartiments Amazon S3

Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

Pour ce didacticiel, créez quelques compartiments à récupérer ultérieurement dans une liste.

Étape 4 : installez le AWS CLI

Installez les instructions AWS CLI suivantes pour votre système d'exploitation. Pour de plus amples informations, veuillez consulter the section called “Installation/mise à jour”.

Une fois l'installation terminée, vous pouvez vérifier l'installation en ouvrant le terminal de votre choix et en exécutant la commande suivante. Cela devrait afficher la version installée du AWS CLI.

```
$ aws --version
```

Étape 5 : Configuration de votre AWS CLI profil

Configurez votre profil à l'aide de l'une des méthodes suivantes

Configurez votre profil avec l'**`aws configure sso`**assistant

La `sso-session` section du `config` fichier est utilisée pour regrouper les variables de configuration permettant d'acquérir des jetons d'accès SSO, qui peuvent ensuite être utilisés pour acquérir des AWS informations d'identification. Les paramètres suivants sont utilisés :

- (Obligatoire) `sso_start_url`
- (Obligatoire) `sso_region`
- `sso_account_id`
- `sso_role_name`
- `sso_registration_scopes`

Vous définissez une `sso-session` section et vous l'associez à un profil. Les `sso_start_url` paramètres `sso_region` et doivent être définis dans la `sso-session` section. Généralement, `sso_account_id` et `sso_role_name` doit être défini dans la `profile` section afin que le SDK puisse demander des informations d'identification SSO.

L'exemple suivant configure le SDK pour demander des informations d'identification SSO et prend en charge l'actualisation automatique des jetons :

```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
```

```
SSO registration scopes [None]: sso:account:access
```

L'autorisation PKCE (Proof Key for Code Exchange) est utilisée par défaut à AWS CLI partir de la version 2.22.0 et doit être utilisée sur les appareils dotés d'un navigateur. Pour continuer à utiliser l'autorisation de l'appareil, ajoutez l'`--use-device-code`option.

```
$ aws configure sso --use-device-code
```

Configuration manuelle à l'aide du **`config`** fichier

La `sso-session` section du `config` fichier est utilisée pour regrouper les variables de configuration permettant d'acquérir des jetons d'accès SSO, qui peuvent ensuite être utilisés pour acquérir des AWS informations d'identification. Les paramètres suivants sont utilisés :

- (Obligatoire) `sso_start_url`
- (Obligatoire) `sso_region`
- `sso_account_id`
- `sso_role_name`
- `sso_registration_scopes`

Vous définissez une `sso-session` section et vous l'associez à un profil. `sso_region`et `sso_start_url` doit être défini dans la `sso-session` section. Généralement, `sso_account_id` et `sso_role_name` doit être défini dans la `profile` section afin que le SDK puisse demander des informations d'identification SSO.

L'exemple suivant configure le SDK pour demander des informations d'identification SSO et prend en charge l'actualisation automatique des jetons :

```
[profile my-dev-profile]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Le jeton d'authentification est mis en cache sur le disque sous le `~/.aws/sso/cache` répertoire avec un nom de fichier basé sur le nom de session.

Étape 6 : Connectez-vous à IAM Identity Center

> **Note**
>
> Le processus de connexion peut vous demander d'autoriser l' AWS CLI accès à vos données. Étant donné que le AWS CLI est construit au-dessus du SDK pour Python, les messages d'autorisation peuvent contenir des variantes du `botocore` nom.

Pour récupérer et mettre en cache vos informations d'identification IAM Identity Center, exécutez la commande suivante pour AWS CLI ouvrir votre navigateur par défaut et vérifier votre connexion à IAM Identity Center.

```
$ aws sso login --profile my-dev-profile
```

À partir de la version 2.22.0, l'autorisation PKCE est la valeur par défaut. Pour utiliser l'autorisation de l'appareil pour vous connecter, ajoutez l'`--use-device-code`option.

```
$ aws sso login --profile my-dev-profile --use-device-code
```

Étape 7 : Exécuter les commandes Amazon S3

Pour répertorier les compartiments que vous avez créés précédemment, utilisez la `aws s3 ls`commande. L'exemple suivant répertorie tous vos compartiments Amazon S3.

```
$ aws s3 ls
2018-12-11 17:08:50 my-bucket
2018-12-14 14:55:44 my-bucket2
```

Étape 8 : Déconnectez-vous d'IAM Identity Center

Lorsque vous avez terminé d'utiliser votre profil IAM Identity Center, exécutez la commande suivante pour supprimer vos informations d'identification mises en cache.

```
$ aws sso logout
```

```
Successfully signed out of all SSO profiles.
```

Étape 9 : Nettoyer les ressources

Une fois ce didacticiel terminé, nettoyez toutes les ressources que vous avez créées au cours de ce didacticiel dont vous n'avez plus besoin, y compris les compartiments Amazon S3.

Résolution des problèmes

Si vous rencontrez des problèmes lors de l'utilisation du AWS CLI, consultez the section called “Résoudre les erreurs” les étapes de dépannage les plus courantes.

Ressources supplémentaires

Les ressources supplémentaires sont les suivantes.

- the section called “Concepts du centre d'identité IAM”
- the section called “Authentification IAM Identity Center”
- the section called “Installation/mise à jour”
- the section called “Paramètres de configuration”
- `aws configure sso`dans la AWS CLI version 2 Reference
- `aws configure sso-session`dans la AWS CLI version 2 Reference
- `aws sso login`dans la AWS CLI version 2 Reference
- `aws sso logout`dans la AWS CLI version 2 Reference
- Configuration pour utiliser le AWS CLI with CodeCatalyst dans le guide de CodeCatalyst l'utilisateur Amazon
- OAuth Étendue d'accès 2.0 dans le guide de l'utilisateur d'IAM Identity Center
- Tutoriels de mise en route dans le guide de l'utilisateur d'IAM Identity Center

Authentification à l'aide d'informations d'identification à court terme pour AWS CLI

Nous vous recommandons de configurer votre SDK ou votre outil pour utiliser l'authentification IAM Identity Center avec des options de durée de session prolongée. Toutefois, vous pouvez copier et utiliser les informations d'identification temporaires disponibles sur le portail AWS d'accès. Les nouvelles informations d'identification devront être copiées à leur expiration. Vous pouvez utiliser

les informations d'identification temporaires dans un profil ou les utiliser comme valeurs pour les propriétés du système et les variables d'environnement.

1. Connectez-vous au portail d' AWS accès.
2. Suivez ces instructions pour copier les informations d'identification du rôle IAM depuis le portail AWS d'accès.

    1. Pour l'étape 2 des instructions associées, choisissez le AWS compte et le nom du rôle IAM qui accordent l'accès pour vos besoins de développement. Ce rôle porte généralement un nom tel que PowerUserAccess« Développeur ».
    2. Pour l'étape 4, sélectionnez l'option Ajouter un profil à votre fichier AWS d'informations d'identification et copiez le contenu.
3. Créez ou ouvrez le `credentials` fichier partagé. Ce fichier se trouve `~/.aws/credentials` sur les systèmes Linux et macOS, ainsi que `%USERPROFILE%\.aws\credentials` sur Windows. Pour de plus amples informations, veuillez consulter the section called “Paramètres de configuration”.
4. Ajoutez le texte suivant au `credentials` fichier partagé. Remplacez les valeurs d'échantillon par les informations d'identification que vous avez copiées.

    ```
    [default]
    aws_access_key_id = AKIAIOSFODNN7EXAMPLE
    aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
    aws_session_token =
     IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
    ```
5. Ajoutez la région et le format par défaut de votre choix au `config` fichier partagé.

    ```
    [default]
    region=us-west-2
    output=json

    [profile user1]
    region=us-east-1
    output=text
    ```

Lorsque le SDK crée un client de service, il accède à ces informations d'identification temporaires et les utilise pour chaque demande. Les paramètres du rôle IAM choisi à l'étape 2a déterminent la durée de validité des informations d'identification temporaires. La durée maximale est de douze heures.

Répétez ces étapes chaque fois que vos informations d'identification expirent.

Utilisation d'un rôle IAM dans AWS CLI

Un rôle AWS Identity and Access Management (IAM) est un outil d'autorisation qui permet à un utilisateur d'obtenir des autorisations supplémentaires (ou différentes), ou d'obtenir des autorisations pour effectuer des actions dans un autre AWS compte.

Rubriques

- Prérequis
- Vue d'ensemble de l'utilisation des rôles IAM
- Configuration et utilisation d'un rôle
- Utilisation de l'authentification multi-facteurs
- Rôles entre comptes et ID externe
- Spécification d'un nom de session de rôle pour faciliter l'audit
- Utilisation d'un rôle à l'aide d'une identité web
- Suppression des informations d'identification mises en cache

Prérequis

Pour exécuter les `iam` commandes, vous devez installer et configurer le AWS CLI. Cela inclut la configuration d'un profil configuré, en supposant qu'un rôle soit associé à une autre méthode d'identification. Pour de plus amples informations, veuillez consulter the section called “Installation/mise à jour”.

Vue d'ensemble de l'utilisation des rôles IAM

Vous pouvez configurer le AWS Command Line Interface (AWS CLI) pour utiliser un rôle IAM en définissant un profil pour le rôle dans le `~/.aws/config` fichier.

L'exemple suivant illustre un profil de rôle nommé `marketingadmin`. Si vous exécutez des commandes avec `--profile marketingadmin` (ou si vous le spécifiez avec la variable d'AWS_PROFILE environnement), AWS CLI utilise les informations d'identification définies dans un profil distinct `user1` pour assumer le rôle avec l'Amazon Resource Name

(ARN)`arn:aws:iam::123456789012:role/marketingadminrole`. Vous pouvez exécuter toutes les opérations permises par les autorisations affectées à ce rôle.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
source_profile = user1
```

Vous pouvez ensuite spécifier un `source_profile` qui pointe vers un profil nommé distinct contenant les informations d'identification des utilisateurs autorisés à utiliser le rôle. Dans l'exemple précédent, le profil `marketingadmin` utilise les informations d'identification du profil `user1`. Lorsque vous spécifiez qu'une AWS CLI commande doit utiliser le profil`marketingadmin`, elle recherche AWS CLI automatiquement les informations d'identification du `user1` profil lié et les utilise pour demander des informations d'identification temporaires pour le rôle IAM spécifié. Pour ce faire, la CLI utilise l'AssumeRoleopération sts : en arrière-plan. Ces informations d'identification temporaires sont ensuite utilisées pour exécuter la AWS CLI commande demandée. Le rôle spécifié doit être associé à des politiques d'autorisation IAM autorisant l'exécution de la AWS CLI commande demandée.

Pour exécuter une AWS CLI commande depuis une instance Amazon Elastic Compute Cloud (Amazon EC2) ou un conteneur Amazon Elastic Container Service (Amazon ECS), vous pouvez utiliser un rôle IAM associé au profil d'instance ou au conteneur. Si vous ne spécifiez aucun profil ou ne définissez aucune variable d'environnement, ce rôle est utilisé directement. Cela vous permet d'éviter de stocker des clés d'accès à longue durée de vie sur vos instances. Vous pouvez également utiliser ces rôles d'instance ou de conteneur uniquement pour obtenir les informations d'identification d'un autre rôle. Pour ce faire, vous devez utiliser `credential_source` (au lieu de `source_profile`) pour spécifier comment trouver les informations d'identification. L'attribut `credential_source` prend en charge les valeurs suivantes :

- `Environment`— Récupère les informations d'identification de la source à partir des variables d'environnement.
- `Ec2InstanceMetadata`— Utilise le rôle IAM associé au profil d' EC2instance Amazon.
- `EcsContainer`— Utilise le rôle IAM associé au conteneur Amazon ECS.

L'exemple suivant montre le même `marketingadminrole` rôle que celui utilisé pour référencer un profil d' EC2 instance Amazon.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
```

```
credential_source = Ec2InstanceMetadata
```

Lorsque vous appelez un rôle, vous disposez d'options supplémentaires dont vous pouvez avoir besoin, telles que l'utilisation d'une authentification multifacteur et d'un ID externe (utilisé par des sociétés tierces pour accéder aux ressources de leurs clients). Vous pouvez également spécifier des noms de session de rôle uniques qui peuvent être audités plus facilement dans AWS CloudTrail les journaux.

Configuration et utilisation d'un rôle

Lorsque vous exécutez des commandes à l'aide d'un profil qui spécifie un rôle IAM, il AWS CLI utilise les informations d'identification du profil source pour appeler AWS Security Token Service (AWS STS) et demander des informations d'identification temporaires pour le rôle spécifié. L'utilisateur dans le profil source doit avoir l'autorisation d'appeler `sts:assume-role` pour le rôle dans le profil spécifié. Le rôle doit avoir une relation d'approbation qui permet à l'utilisateur dans le profil source d'utiliser le rôle. Le processus de récupération, puis d'utilisation d'informations d'identification temporaires pour un rôle revient à assumer le rôle.

Vous pouvez créer un rôle dans IAM avec les autorisations que vous souhaitez que les utilisateurs assument en suivant la procédure décrite dans la section Création d'un rôle pour déléguer des autorisations à un utilisateur IAM dans le Guide de l'AWS Identity and Access Management utilisateur. Si le rôle et l'utilisateur du profil source sont dans le même compte, vous pouvez entrer votre propre ID de compte lors de la configuration de la relation de confiance du rôle.

Une fois le rôle créé, modifiez la relation de confiance afin d'autoriser l'utilisateur à assumer ce rôle.

L'exemple suivant illustre une stratégie d'approbation que vous pouvez attacher à un rôle. Cette politique permet que le rôle soit assumé par n'importe quel utilisateur du compte 123456789012, si l'administrateur de ce compte accorde explicitement l'autorisation à l'`sts:AssumeRole`utilisateur.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
```

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

La stratégie d'approbation n'accorde pas actuellement d'autorisations. L'administrateur du compte doit déléguer l'autorisation d'assumer le rôle à des utilisateurs individuels en associant une stratégie aux autorisations appropriées. L'exemple suivant montre une politique que vous pouvez associer à un utilisateur et qui permet à ce dernier d'assumer uniquement le `marketingadminrole` rôle. Pour plus d'informations sur l'octroi à un utilisateur de l'accès lui permettant d'assumer un rôle, consultez la section Octroi à un utilisateur de l'autorisation de changer de rôle dans le guide de l'utilisateur IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

L'utilisateur n'a pas besoin d'autorisations supplémentaires pour exécuter les AWS CLI commandes à l'aide du profil de rôle. Au lieu de cela, les autorisations pour exécuter la commande proviennent de celles qui sont attachées au rôle. Vous associez des politiques d'autorisation au rôle pour spécifier quelles actions peuvent être effectuées sur quelles AWS ressources. Pour plus d'informations sur l'attribution d'autorisations à un rôle (qui fonctionne de la même manière que pour un utilisateur), consultez la section Modification des autorisations pour un utilisateur IAM dans le guide de l'utilisateur IAM.

Maintenant que vous avez correctement configuré le profil de rôle, les autorisations de rôle, la relation d'approbation du rôle et les autorisations utilisateur, vous pouvez utiliser ce rôle au niveau de la ligne de commande en appelant l'option `--profile`. Par exemple, ce qui suit appelle la `ls` commande

Amazon S3 en utilisant les autorisations associées au `marketingadmin` rôle, telles que définies dans l'exemple au début de cette rubrique.

```
$ aws s3 ls --profile marketingadmin
```

Si vous souhaitez utiliser le rôle pour plusieurs appels, vous pouvez définir la variable d'environnement `AWS_PROFILE` pour la session en cours à partir de la ligne de commande. Lorsque cette variable d'environnement est définie, vous n'avez pas besoin de spécifier l'option `--profile` pour chaque commande.

Linux ou macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

Pour plus d'informations sur la configuration des utilisateurs et des rôles, consultez les sections Identités IAM (utilisateurs, groupes d'utilisateurs et rôles) et rôles IAM dans le guide de l'utilisateur IAM.

Utilisation de l'authentification multi-facteurs

Pour une sécurité accrue, vous pouvez exiger que les utilisateurs fournissent une clé à usage unique générée à partir d'un périphérique d'authentification multi-facteurs (MFA), d'un appareil U2F ou d'une application mobile lorsqu'ils essaient d'effectuer un appel en utilisant le profil de rôle.

Tout d'abord, vous pouvez choisir de modifier la relation de confiance sur le rôle IAM pour exiger l'authentification MFA. De cette façon, aucun utilisateur ne peut utiliser le rôle sans l'authentification MFA. Pour obtenir un exemple, consultez la ligne `Condition` dans l'exemple suivant. Cette politique permet à l'utilisateur nommé d'`anika`assumer le rôle auquel la politique est attachée, mais uniquement s'il s'authentifie à l'aide de la MFA.

JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/anika" },
      "Action": "sts:AssumeRole",
      "Condition": { "Bool": { "aws:multifactorAuthPresent": true } }
    }
  ]
}
```

Ajoutez ensuite une ligne au profil de rôle qui spécifie l'ARN du périphérique MFA de l'utilisateur. Les exemples d'entrées de `config` fichier suivants présentent deux profils de rôle qui utilisent tous deux les clés d'accès permettant `anika` à l'utilisateur de demander des informations d'identification temporaires pour le rôle`cli-role`. L'utilisateur `anika` dispose des autorisations requises pour assumer le rôle, accordées par la stratégie d'approbation du rôle.

```
[profile role-without-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile=cli-user

[profile role-with-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile = cli-user
mfa_serial = arn:aws:iam::128716708097:mfa/cli-user

[profile cli-user]
region = us-west-2
output = json
```

Le paramètre `mfa_serial` peut utiliser un ARN, comme indiqué, ou le numéro de série d'un jeton MFA matériel.

Le premier profil, `role-without-mfa`, ne nécessite pas d'authentification MFA. Cependant, étant donné que l'exemple précédent de stratégie d'approbation attachée au rôle exige l'authentification MFA, toute tentative d'exécuter une commande avec ce profil échoue.

```
$ aws iam list-users --profile role-without-mfa
```

```
An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

La deuxième entrée de profil, `role-with-mfa`, identifie un périphérique MFA à utiliser. Lorsque l'utilisateur tente d'exécuter une AWS CLI commande avec ce profil, il est AWS CLI invité à saisir le mot de passe à usage unique (OTP) fourni par le dispositif MFA. Si l'authentification MFA réussit, la commande exécute l'opération demandée. L'OTP ne s'affiche pas à l'écran.

```
$ aws iam list-users --profile role-with-mfa
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
{
    "Users": [
        {
            ...
```

Rôles entre comptes et ID externe

Vous pouvez permettre aux utilisateurs d'utiliser des rôles qui appartiennent à différents comptes en configurant le rôle en tant que rôle entre comptes. Lors de la création du rôle, définissez le type de rôle sur Autre AWS compte, comme décrit dans Création d'un rôle pour déléguer des autorisations à un utilisateur IAM. Le cas échéant, sélectionnez MFA obligatoire. L'option MFA obligatoire configure la condition appropriée dans la relation de confiance comme décrit dans la section Utilisation de l'authentification multi-facteurs.

Si vous utilisez un ID externe pour accorder un contrôle supplémentaire aux personnes susceptibles d'utiliser un rôle entre les comptes, vous devez ajouter le paramètre `external_id` au profil de rôle. Généralement, vous utilisez uniquement cette option lorsque l'autre compte est contrôlé par une personne qui ne fait pas partie de votre entreprise ou organisation.

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

Spécification d'un nom de session de rôle pour faciliter l'audit

Lorsque de nombreuses personnes partagent un rôle, la vérification devient plus difficile. Vous souhaitez associer chaque opération appelée à la personne à l'origine de cet appel. Toutefois,

lorsque la personne utilise un rôle, la prise en charge du rôle est une action distincte de l'appel d'une opération, et vous devez corréler les deux manuellement.

Vous pouvez simplifier cela en spécifiant des noms de session de rôle uniques lorsque les utilisateurs assument un rôle. Pour ce faire, vous devez ajouter un paramètre `role_session_name` à chaque profil nommé dans le fichier `config` qui spécifie un rôle. La valeur de `role_session_name` est transmise à l'opération `AssumeRole` et devient partie intégrante de l'ARN de la session de rôle. Il est également inclus dans les AWS CloudTrail journaux de toutes les opérations enregistrées.

Par exemple, vous pouvez créer un profil basé sur le rôle comme suit :

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
role_session_name = Session_Maria_Garcia
```

La session de rôle prend alors l'ARN suivant :

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

En outre, tous les AWS CloudTrail journaux incluent le nom de session du rôle dans les informations capturées pour chaque opération.

Utilisation d'un rôle à l'aide d'une identité web

Vous pouvez configurer un profil pour indiquer qu'il AWS CLI doit assumer un rôle à l'aide de la fédération d'identité Web et d'Open ID Connect (OIDC). Lorsque vous le spécifiez dans un profil, il AWS CLI passe automatiquement l' AWS STS `AssumeRoleWithWebIdentity`appel correspondant pour vous.

> **Note**
>
> Lorsque vous spécifiez un profil qui utilise un rôle IAM, il AWS CLI effectue les appels appropriés pour récupérer des informations d'identification temporaires. Ces informations d'identification sont stockées dans `~/.aws/cli/cache`. Les AWS CLI commandes suivantes qui spécifient le même profil utilisent les informations d'identification temporaires mises en cache jusqu'à leur expiration. À ce stade, les informations d'identification AWS CLI sont automatiquement actualisées.

Pour récupérer et utiliser des informations d'identification temporaires à l'aide de la fédération d'identité web, vous pouvez spécifier les valeurs de configuration suivantes dans un profil partagé :

role_arn

Spécifie l'ARN du rôle à assumer.

web_identity_token_file

Spécifie le chemin d'accès à un fichier contenant un jeton d'accès OAuth 2.0 ou un jeton d'identification OpenID Connect fourni par le fournisseur d'identité. L' AWS CLI charge ce fichier et transmet son contenu en tant qu'argument `WebIdentityToken` de l'opération `AssumeRoleWithWebIdentity`.

role_session_name

Spécifie un nom facultatif appliqué à cette session assume-role.

Voici un exemple de configuration pour la configuration minimale requise pour configurer un profil de rôle de responsable avec profil identité web :

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoleNameToAssume
web_identity_token_file=/path/to/a/token
```

Vous pouvez également fournir cette configuration à l'aide de variables d'environnement :

AWS_ROLE_ARN

ARN du rôle à assumer.

AWS_WEB_FICHIER_JETON D'IDENTITÉ

Chemin d'accès au fichier de jeton d'identité web.

AWS_ROLE_NOM_SESSION

Nom appliqué à cette session assume-role.

> **Note**
>
> Ces variables d'environnement s'appliquent actuellement uniquement au rôle « assume » avec le fournisseur d'identité web. Ils ne s'appliquent pas à la configuration générale du fournisseur de rôle.

Suppression des informations d'identification mises en cache

Lorsque vous utilisez un rôle, les informations d'identification temporaires sont mises en AWS CLI cache localement jusqu'à leur expiration. La prochaine fois que vous essaierez de les utiliser, ils AWS CLI essaieront de les renouveler en votre nom.

Si les informations d'identification temporaires de votre rôle sont révoquées, elles ne sont pas renouvelées automatiquement et les tentatives d'utilisation échouent. Cependant, vous pouvez supprimer le cache pour forcer le AWS CLI à récupérer de nouvelles informations d'identification.

Linux ou macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

Authentification à l'aide des informations d'identification utilisateur IAM pour AWS CLI

> **Warning**
>
> Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que AWS IAM Identity Center.

Cette section explique comment configurer les paramètres de base avec un utilisateur IAM. Il s'agit notamment de vos informations de sécurité à l'aide `config` des `credentials` fichiers et. Pour

consulter à la place les instructions de configuration pour AWS IAM Identity Center, voirthe section called “Authentification IAM Identity Center”.

Rubriques

- Étape 1 : créer votre utilisateur IAM
- Étape 2 : Obtenez vos clés d'accès
- Configurez le AWS CLI
  - Utiliser aws configure
  - Importation de clés d'accès via un fichier .CSV
  - Modification directe des credentials fichiers config et

Étape 1 : créer votre utilisateur IAM

Créez votre utilisateur IAM en suivant la procédure de création d'utilisateurs IAM (console) du guide de l'utilisateur IAM.

- Pour les options d'autorisation, choisissez Joindre directement les politiques pour définir la manière dont vous souhaitez attribuer les autorisations à cet utilisateur.
- La plupart des didacticiels du SDK « Getting Started » utilisent le service Amazon S3 comme exemple. Pour fournir à votre application un accès complet à Amazon S3, sélectionnez la `AmazonS3FullAccess` politique à associer à cet utilisateur.

Étape 2 : Obtenez vos clés d'accès

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
2. Dans le volet de navigation de la console IAM, sélectionnez Utilisateurs, puis sélectionnez **`User name`**l'utilisateur que vous avez créé précédemment.
3. Sur la page de l'utilisateur, sélectionnez la page Informations d'identification de sécurité. Ensuite, sous Clés d'accès, sélectionnez Créer une clé d'accès.
4. Pour l'étape 1 de création d'une clé d'accès, choisissez Command Line Interface (CLI).
5. Pour l'étape 2 de création d'une clé d'accès, entrez une balise facultative et sélectionnez Suivant.

6. Pour l'étape 3 de création d'une clé d'accès, sélectionnez Télécharger le fichier .csv pour enregistrer un `.csv` fichier avec la clé d'accès et la clé d'accès secrète de votre utilisateur IAM. Vous avez besoin de ces informations pour plus tard.
7. Sélectionnez Done (Terminé).

Configurez le AWS CLI

Pour une utilisation générale, les AWS CLI informations suivantes sont nécessaires :

- ID de clé d'accès
- Clé d'accès secrète
- AWS Région
- Format de sortie

AWS CLI Stocke ces informations dans un profil (un ensemble de paramètres) nommé `default` dans le `credentials` fichier. Par défaut, les informations de ce profil sont utilisées lorsque vous exécutez une AWS CLI commande qui ne spécifie pas explicitement le profil à utiliser. Pour de plus amples informations sur le fichier `credentials`, veuillez consulter Paramètres des fichiers de configuration et d'identification dans le AWS CLI.

Pour configurer le AWS CLI, appliquez l'une des procédures suivantes :

Rubriques

- Utiliser aws configure
- Importation de clés d'accès via un fichier .CSV
- Modification directe des credentials fichiers config et

Utiliser **`aws configure`**

Pour une utilisation générale, la `aws configure` commande est le moyen le plus rapide de configurer votre AWS CLI installation. Cet assistant de configuration vous invite à saisir toutes les informations dont vous avez besoin pour commencer. Sauf indication contraire à l'aide de l'`--profile`option, cette information est AWS CLI stockée dans le `default` profil.

L'exemple suivant configure un `default` profil à l'aide d'exemples de valeurs. Remplacez-les par vos propres valeurs comme décrit dans les sections suivantes.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

L'exemple suivant configure un profil nommé à `userprod` l'aide d'exemples de valeurs. Remplacez-les par vos propres valeurs comme décrit dans les sections suivantes.

```
$ aws configure --profile userprod
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Importation de clés d'accès via un fichier .CSV

Au lieu de `aws configure` saisir les clés d'accès, vous pouvez importer le `.csv` fichier texte brut que vous avez téléchargé après avoir créé vos clés d'accès.

Le `.csv` fichier doit contenir les en-têtes suivants.

- Nom d'utilisateur - Cette colonne doit être ajoutée à votre`.csv`. Ceci est utilisé pour créer le nom de profil utilisé dans les `credentials` fichiers `config` et lors de l'importation.
- ID de clé d'accès
- Clé d'accès secrète

> **Note**
>
> Lors de la création initiale des clés d'accès, une fois que vous avez fermé la boîte de dialogue Télécharger le fichier .csv, vous ne pouvez plus accéder à votre clé d'accès secrète après avoir fermé la boîte de dialogue. Si vous avez besoin d'un `.csv` fichier, vous devez en créer un vous-même avec les en-têtes requis et les informations relatives à vos clés d'accès enregistrées. Si vous n'avez pas accès aux informations relatives à vos clés d'accès, vous devez créer de nouvelles clés d'accès.

Pour importer le `.csv` fichier, utilisez la `aws configure import` commande avec l'`--csv`option suivante :

```
$ aws configure import --csv file://credentials.csv
```

Pour de plus amples informations, veuillez consulter `aws_configure_import`.

Modification directe des **`credentials`** fichiers **`config`** et

Pour modifier directement les `credentials` fichiers `config` et, effectuez les opérations suivantes.

1. Créez ou ouvrez le AWS `credentials` fichier partagé. Ce fichier se trouve `~/.aws/credentials` sur les systèmes Linux et macOS, ainsi que `%USERPROFILE%\.aws\credentials` sur Windows. Pour de plus amples informations, veuillez consulter the section called “Paramètres de configuration”.
2. Ajoutez le texte suivant au `credentials` fichier partagé. Remplacez les valeurs d'exemple dans le `.csv` fichier que vous avez téléchargé précédemment et enregistrez le fichier.

   ```
   [default]
   aws_access_key_id = AKIAIOSFODNN7EXAMPLE
   aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   ```

Utilisation des métadonnées de l' EC2 instance Amazon comme informations d'identification dans AWS CLI

Lorsque vous exécutez le formulaire AWS CLI depuis une instance Amazon Elastic Compute Cloud (Amazon EC2), vous pouvez simplifier la fourniture d'informations d'identification pour vos commandes. Chaque EC2 instance Amazon contient des métadonnées qui AWS CLI permettent de demander directement des informations d'identification temporaires. Lorsqu'un rôle IAM est attaché à l'instance, il récupère AWS CLI automatiquement et en toute sécurité les informations d'identification à partir des métadonnées de l'instance.

Pour désactiver ce service, utilisez la variable d'environnement AWS_EC2_METADATA_DISABLED.

Rubriques

- Prérequis
- Configuration d'un profil pour les EC2 métadonnées Amazon

Prérequis

Pour utiliser les EC2 informations d'identification Amazon avec le AWS CLI, vous devez effectuer les opérations suivantes :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Vous comprenez les fichiers de configuration et les profils nommés. Pour de plus amples informations, veuillez consulter Paramètres des fichiers de configuration et d'identification dans le AWS CLI.
- Vous avez créé un rôle AWS Identity and Access Management (IAM) qui a accès aux ressources nécessaires, et vous avez attaché ce rôle à l' EC2 instance Amazon lorsque vous la lancez. Pour plus d'informations, consultez les politiques IAM pour Amazon EC2 dans le guide de l' EC2 utilisateur Amazon et l'octroi de l'accès aux AWS ressources aux applications exécutées sur EC2 des instances Amazon dans le guide de l'utilisateur IAM.

Configuration d'un profil pour les EC2 métadonnées Amazon

Pour spécifier que vous souhaitez utiliser les informations d'identification disponibles dans le profil d' EC2instance Amazon d'hébergement, utilisez la syntaxe suivante dans le profil nommé de votre fichier de configuration. Consultez les étapes suivantes pour plus d'instructions.

```
[profile profilename]
role_arn = arn:aws:iam::123456789012:role/rolename
credential_source = Ec2InstanceMetadata
region = region
```

1. Créez un profil dans votre fichier de configuration.

   ```
   [profile profilename]
   ```

2. Ajoutez votre rôle IAM arn qui a accès aux ressources nécessaires.

   ```
   role_arn = arn:aws:iam::123456789012:role/rolename
   ```

3. Spécifiez `Ec2InstanceMetadata` comme source d'informations d'identification.

   ```
   credential_source = Ec2InstanceMetadata
   ```

4. Définissez votre région.

```
region = region
```

Exemple

L'exemple suivant assume le *marketingadminrole* rôle et utilise la *us-west-2* région dans un profil d' EC2 instance Amazon nommé*marketingadmin*.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
region = us-west-2
```

Obtenir des informations d'identification par le biais d'un processus externe dans le AWS CLI

> **Warning**
>
> La rubrique aborde l'utilisation d'informations d'identification provenant d'un processus d'approvisionnement externe. Un risque de sécurité peut exister si la commande de génération des informations d'identification devient accessible à des processus ou utilisateurs non approuvés. Nous vous recommandons d'utiliser les alternatives sécurisées prises en charge par le AWS CLI et AWS afin de réduire le risque de compromettre vos informations d'identification. Veillez à sécuriser le fichier `config` et tous les outils et fichiers de support afin d'éviter toute divulgation.
> Assurez-vous que votre outil d'identification personnalisé n'écrit aucune information secrète, `StdErr` car le SDKs et AWS CLI peut capturer et enregistrer ces informations, les exposant potentiellement à des utilisateurs non autorisés.

Si vous disposez d'une méthode pour générer ou rechercher des informations d'identification qui n'est pas directement prise en charge par le AWS CLI, vous pouvez la configurer AWS CLI pour l'utiliser en configurant le `credential_process` paramètre dans le `config` fichier.

Par exemple, vous pourriez inclure une entrée comme celle-ci dans le fichier `config` :

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

Syntaxe

Pour créer cette chaîne d'une manière compatible avec n'importe quel système d'exploitation, suivez ces règles :

- Si le chemin d'accès ou le nom du fichier contient un espace, entourez le chemin d'accès complet et le nom du fichier de guillemets doubles (" "). Le chemin et le nom de fichier peuvent contenir uniquement les caractères suivants : A-Z a-z 0-9 - _. espace
- Si un nom de paramètre ou une valeur de paramètre contient un espace, entourez cet élément de guillemets doubles (" "). Entourez uniquement le nom ou la valeur, pas la paire.
- N'incluez aucune variable d'environnement dans les chaînes. Par exemple, vous ne pouvez pas inclure `$HOME` ou `%USERPROFILE%`.
- Ne spécifiez pas le dossier de base comme ~. Vous devez spécifier le chemin d'accès complet.

Exemple pour Windows

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter
 with spaces"
```

Exemple pour Linux ou macOS

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces
 "parameter with spaces"
```

Sortie attendue du programme d'informations d'identification

AWS CLI Exécute la commande comme indiqué dans le profil, puis lit les données à partir de`STDOUT`. La commande que vous spécifiez doit générer sur `STDOUT` une sortie JSON qui correspond à la syntaxe suivante :

```
{
  "Version": 1,
  "AccessKeyId": "an AWS access key",
  "SecretAccessKey": "your AWS secret access key",
  "SessionToken": "the AWS session token for temporary credentials",
```

```
  "Expiration": "ISO8601 timestamp when the credentials expire"
}
```

> **Note**
>
> A la date de cette publication, la clé `Version` doit être définie sur 1. Cela pourrait augmenter avec le temps à mesure que la structure évolue.

La `Expiration` clé est un horodatage au format ISO8601. Si la clé `Expiration` n'est pas présente dans la sortie de l'outil, l'interface de ligne de commande suppose que les informations d'identification sont des informations d'identification à long terme qui ne s'actualisent pas. Dans le cas contraire, les informations d'identification sont considérées comme temporaires et sont actualisées automatiquement en réexécutant la commande `credential_process` avant leur expiration.

> **Note**
>
> Le AWS CLI ne met pas en cache les informations d'identification des processus externes de la même manière qu'il assume les informations d'identification des rôles. Si la mise en cache est obligatoire, vous devez la mettre en œuvre dans le processus externe.

Le processus externe peut renvoyer un code de retour non nul pour indiquer qu'une erreur s'est produite lors de la récupération des informations d'identification.

En utilisant le AWS CLI

Cette section fournit un aperçu complet de l'utilisation générale, des fonctionnalités communes et des options disponibles dans le AWS Command Line Interface (AWS CLI), allant au-delà des détails abordés dans la the section called “Points de terminaison” section Configuration.

Ce guide aborde les aspects fondamentaux de l'écriture de AWS CLI commandes, notamment leur structure de base, leur formatage et leurs capacités de filtrage. En comprenant ces éléments essentiels, vous serez en mesure de créer des commandes qui ciblent précisément les ressources et les actions dont vous avez besoin, sans avoir à naviguer dans des consoles Web complexes.

Cela met également en évidence le contenu d'aide et la documentation disponibles pour le AWS CLI. De l'aide en ligne de commande intégrée au complet de la AWS CLI version 2 du guide de référence, vous aurez accès à des informations qui vous aideront à explorer les fonctionnalités et les capacités du AWS CLI.

Pour Service AWS des exemples et des cas d'utilisation spécifiques, consultez AWS CLI exemples le version 2 du guide de référence. Ils fournissent des informations spécifiques aux commandes et montrent des exemples sur la manière d'en tirer parti AWS CLI pour différents Services AWS.

> **Note**
>
> Par défaut, les demandes sont AWS CLI envoyées à l'aide Services AWS du protocole HTTPS sur le port TCP 443. Pour garantir une utilisation réussie du AWS CLI, vous devez être en mesure d'établir des connexions sortantes sur ce port.

Rubriques de ce guide

- Accès à l'aide et aux ressources pour AWS CLI
- Structure de commande dans le AWS CLI
- Spécifier les valeurs des paramètres dans le AWS CLI
- Activation et utilisation des invites de commande dans le AWS CLI
- Contrôle de la sortie des commandes dans le AWS CLI
- Codes de retour de ligne de commande dans le AWS CLI
- Utilisation d'assistants personnalisés pour exécuter des commandes interactives dans AWS CLI

- Création et utilisation d'alias dans AWS CLI
- Résolution des erreurs liées au AWS CLI

Accès à l'aide et aux ressources pour AWS CLI

Cette rubrique décrit comment accéder au contenu d'aide pour le AWS Command Line Interface (AWS CLI).

Rubriques

- La commande d' AWS CLI aide intégrée
- AWS CLI guide de référence
- Documentation sur les API
- Résolution des erreurs
- Aide supplémentaire

La commande d' AWS CLI aide intégrée

Vous pouvez obtenir de l'aide avec n'importe quelle commande lorsque vous utilisez le AWS Command Line Interface (AWS CLI). Pour cela, il vous suffit de saisir `help` à la fin d'un nom de commande.

Par exemple, la commande suivante affiche de l'aide concernant les AWS CLI options générales et les commandes de niveau supérieur disponibles.

```
$ aws help
```

La commande suivante affiche les commandes spécifiques à Amazon Elastic Compute Cloud (Amazon EC2) disponibles.

```
$ aws ec2 help
```

L'exemple suivant affiche une aide détaillée pour le EC2 `DescribeInstances` fonctionnement d'Amazon. L'aide inclut des descriptions de ses paramètres d'entrée, les filtres disponibles, et ce qui est inclus dans la sortie. Il inclut également des exemples illustrant la façon de saisir des variations courantes de la commande.

```
$ aws ec2 describe-instances help
```

L'aide de chaque commande est divisée en six sections :

Nom

Le nom de la commande.

```
NAME
       describe-instances -
```

Description

Une description de l'opération d'API que la commande appelle.

```
DESCRIPTION
       Describes one or more of your instances.

       If you specify one or more instance IDs, Amazon EC2 returns information
       for those instances. If you do not specify  instance  IDs,  Amazon  EC2
       returns  information  for  all  relevant  instances.  If you specify an
       instance ID that is not valid, an error is returned. If you specify  an
       instance  that  you  do  not  own,  it  is not included in the returned
       results.
...
```

Résumé

La syntaxe de base pour utiliser la commande et ses options. Si une option est affichée entre crochets, elle est facultative ou elle a une valeur par défaut, ou bien une autre option peut être utilisée à la place.

```
SYNOPSIS
            describe-instances
          [--dry-run | --no-dry-run]
          [--instance-ids <value>]
          [--filters <value>]
          [--cli-input-json <value>]
          [--starting-token <value>]
          [--page-size <value>]
          [--max-items <value>]
          [--generate-cli-skeleton]
```

Par exemple, `describe-instances` possède un comportement par défaut qui décrit toutes les instances du compte courant et de AWS la région. Si vous le souhaitez, vous pouvez spécifier une liste d'`instance-ids` pour décrire une ou plusieurs instances. `dry-run` est un indicateur booléen facultatif qui ne prend aucune valeur. Pour utiliser un indicateur booléen, indiquez la valeur affichée, dans le cas présent `--dry-run` ou `--no-dry-run`. De même, `--generate-cli-skeleton`on ne prend pas de valeur. S'il existe des conditions sur l'utilisation d'une option, elles sont décrites dans la section `OPTIONS`, ou mentionnées dans les exemples.

Options

Une description de chacune des options affichées dans le résumé.

```
OPTIONS
       --dry-run | --no-dry-run (boolean)
          Checks whether you have the required  permissions  for  the  action,
          without actually making the request, and provides an error response.
          If you have the required permissions, the error response is  DryRun-
          Operation . Otherwise, it is UnauthorizedOperation .

       --instance-ids (list)
          One or more instance IDs.

          Default: Describes all your instances.
...
```

Exemples

Des exemples illustrant l'utilisation de la commande et de ses options. Si aucun exemple n'est disponible pour une commande ou un cas d'utilisation dont vous avez besoin, demandez-en un en utilisant le lien de commentaires sur cette page ou dans la référence de AWS CLI commande sur la page d'aide de la commande.

```
    EXAMPLES
    To describe an Amazon EC2 instance

    Command:

    aws ec2 describe-instances --instance-ids i-5203422c

    To describe all instances with the instance type m1.small

    Command:
```

```
    aws ec2 describe-instances --filters "Name=instance-type,Values=m1.small"

    To describe all instances with an Owner tag

    Command:

    aws ec2 describe-instances --filters "Name=tag-key,Values=Owner"
...
```

Sortie

Des descriptions de chaque champ et type de données inclus dans la réponse d' AWS.

Pour `describe-instances`, la sortie est une liste d'objets de réservation, chacun comportant plusieurs champs et des objets contenant des informations sur les instances qui lui sont associées. Ces informations proviennent de la documentation de l'API pour le type de données de réservation utilisé par Amazon EC2.

```
OUTPUT
       Reservations -> (list)
          One or more reservations.

          (structure)
              Describes a reservation.

              ReservationId -> (string)
                 The ID of the reservation.

              OwnerId -> (string)
                 The ID of the AWS account that owns the reservation.

              RequesterId -> (string)
                 The ID of the requester that launched the instances  on  your
                 behalf (for example, AWS Management Console or Auto Scaling).

              Groups -> (list)
                 One or more security groups.

                 (structure)
                     Describes a security group.

                     GroupName -> (string)
```

```
                        The name of the security group.

                    GroupId -> (string)
                        The ID of the security group.

                Instances -> (list)
                   One or more instances.

                   (structure)
                       Describes an instance.

                       InstanceId -> (string)
                          The ID of the instance.

                       ImageId -> (string)
                          The ID of the AMI used to launch the instance.

                       State -> (structure)
                          The current state of the instance.

                          Code -> (integer)
                              The  low  byte represents the state. The high byte
                              is an opaque internal value and should be ignored.
...
```

Lorsque le résultat est AWS CLI rendu au format JSON, il devient un tableau d'objets de réservation, comme dans l'exemple suivant.

```
{
    "Reservations": [
        {
            "OwnerId": "012345678901",
            "ReservationId": "r-4c58f8a0",
            "Groups": [],
            "RequesterId": "012345678901",
            "Instances": [
                {
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "PublicDnsName": "ec2-52-74-16-12.us-
west-2.compute.amazonaws.com",
                    "State": {
```

```
                    "Code": 16,
                    "Name": "running"
                },
...
```

Chaque objet de réservation contient les champs décrivant la réservation et un tableau des objets de l'instance, chacun avec ses propres champs (`PublicDnsName`, par exemple) et des objets (`State`, par exemple) qui le décrivent.

> **Utilisateurs Windows:**
>
> Vous pouvez acheminer (|) la sortie de la commande d'aide vers la commande `more` pour afficher le fichier d'aide une page à la fois. Appuyez sur la barre d'espace ou PgDnpour afficher une plus grande partie du document, puis **q** pour quitter.
>
> ```
> C:\> aws ec2 describe-instances help | more
> ```

AWS CLI guide de référence

Les fichiers d'aide contiennent des liens qui ne peuvent pas être affichés ou accédés depuis la ligne de commande. Vous pouvez consulter ces liens et interagir avec eux en utilisant le guide de de référence AWS CLI version 2. La référence contient également le contenu d'aide pour toutes les AWS CLI commandes. Les descriptions sont présentées pour faciliter la navigation et l'affichage sur les téléphones portables, les tablettes ou les écrans de bureau.

Documentation sur les API

Toutes les commandes du AWS CLI correspondent aux demandes adressées à l'API AWS publique d'un service. Chaque service doté d'une API publique possède une référence d'API qui se trouve sur la page d'accueil du service sur le site Web de AWS documentation. Le contenu d'une référence d'API varie en fonction de la manière dont l'API est construite et du protocole qui est utilisé. En général, une référence d'API contient des informations détaillées sur les actions prises en charge par l'API, les données envoyées vers et depuis le service et les conditions d'erreur que le service peut signaler.

Sections de la documentation d'API

- Actions : informations détaillées sur chaque opération et ses paramètres (y compris les contraintes de longueur ou de contenu et les valeurs par défaut). Elle répertorie les erreurs qui peuvent se produire pour cette action. Chaque opération correspond à une sous-commande du AWS CLI.
- Types de données : informations détaillées sur les structures qu'une commande peut nécessiter en tant que paramètre ou renvoyées en réponse à une demande.
- Paramètres communs : informations détaillées sur les paramètres communs à toutes les actions du service.
- Erreurs courantes : informations détaillées sur les erreurs pouvant être renvoyées par n'importe quelle opération du service.

Le nom et la disponibilité de chaque section peuvent varier en fonction du service.

> **Spécifique au service CLIs**
>
> Certains services ont une CLI distincte qui date d'avant qu'une seule ne soit AWS CLI créée pour fonctionner avec tous les services. Ces documents spécifiques au service CLIs ont une documentation distincte qui est liée à la page de documentation du service. La documentation spécifique au service CLIs ne s'applique pas au. AWS CLI

Résolution des erreurs

Pour obtenir de l'aide sur le diagnostic et la correction des AWS CLI erreurs, consultezthe section called “Résoudre les erreurs”.

Aide supplémentaire

Pour obtenir de l'aide supplémentaire AWS CLI concernant vos problèmes, consultez la AWS CLI communauté sur GitHub.

Structure de commande dans le AWS CLI

Cette rubrique explique comment la commande AWS Command Line Interface (AWS CLI) est structurée et comment utiliser les commandes wait.

Rubriques

- Structure de commande
- Commandes d'attente

Structure de commande

AWS CLI Utilise une structure en plusieurs parties sur la ligne de commande qui doit être spécifiée dans cet ordre :

1. L'appel de base pour le programme `aws`.
2. La commande de niveau supérieur, qui correspond généralement à un AWS service pris en charge par le AWS CLI.
3. La sous-commande qui spécifie l'opération à effectuer.
4. AWS CLI Options ou paramètres généraux requis par l'opération. Vous pouvez les spécifier dans n'importe quel ordre tant qu'ils suivent les trois premières parties. Si un paramètre exclusif est spécifié plusieurs fois, seule la dernière valeur s'applique.

```
$ aws <command> <subcommand> [options and parameters]
```

Les paramètres peuvent prendre différents types de valeurs d'entrée, comme des chiffres, des chaînes, des listes, des mappages et des structures JSON. Ce qui est pris en charge dépend de la commande et de la sous-commande que vous spécifiez.

Exemples

Amazon S3

L'exemple suivant répertorie tous vos compartiments Amazon S3.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

Pour plus d'informations sur les commandes Amazon S3, consultez la référence des AWS CLI commandes.

AWS CloudFormation

L'exemple de commande suivant remplace le nom de la pile cloudformation par. *my-change-set*

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

Pour plus d'informations sur les AWS CloudFormation commandes, reportez-vous à la référence des AWS CLI commandes.

Commandes d'attente

Des `wait` commandes sont disponibles pour certains AWS services. Toute commande utilisée attend `aws wait` généralement qu'une commande soit terminée avant de passer à l'étape suivante. Cela est particulièrement utile pour les commandes partitionnées ou les scripts, car vous pouvez utiliser une commande d'attente pour empêcher de passer aux étapes suivantes en cas d'échec de la commande d'attente.

AWS CLI Utilise une structure en plusieurs parties sur la ligne de commande pour la `wait` commande qui doit être spécifiée dans cet ordre :

1. L'appel de base pour le programme `aws`.
2. La commande de niveau supérieur, qui correspond généralement à un AWS service pris en charge par le AWS CLI.
3. La commande `wait`.
4. La sous-commande qui spécifie l'opération à effectuer.
5. Les options CLI générales ou les paramètres requis par l'opération. Vous pouvez les spécifier dans n'importe quel ordre tant qu'ils suivent les trois premières parties. Si un paramètre exclusif est spécifié plusieurs fois, seule la dernière valeur s'applique.

```
$ aws <command> wait <subcommand> [options and parameters]
```

Les paramètres peuvent prendre différents types de valeurs d'entrée, comme des chiffres, des chaînes, des listes, des mappages et des structures JSON. Ce qui est pris en charge dépend de la commande et de la sous-commande que vous spécifiez.

> **Note**
>
> Tous les services AWS ne prennent pas en charge `wait` les commandes. Consultez le version 2 pour savoir si votre service prend en charge `wait` les commandes.

Exemples

AWS CloudFormation

Les exemples de commande suivants mettent en pause et reprennent uniquement une fois qu'ils ont confirmé que la *my-change-set* modification définie dans la *my-stack* pile est prête à être exécutée.

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

Pour plus d'informations sur les AWS CloudFormation `wait` commandes, reportez-vous à la référence des AWS CLI commandes.

AWS CodeDeploy

Les exemples de commandes suivants sont interrompus jusqu'à ce que le *d-A1B2C3111* déploiement soit terminé avec succès.

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

Pour plus d'informations sur les AWS CodeDeploy `wait` commandes, reportez-vous à la référence des AWS CLI commandes.

Spécifier les valeurs des paramètres dans le AWS CLI

De nombreux paramètres utilisés dans le AWS Command Line Interface (AWS CLI) sont de simples chaînes ou des valeurs numériques, comme le nom de la paire de clés `my-key-pair` dans l'exemple de commande suivant`aws ec2 create-key-pair`.

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

Le formatage de la commande peut varier d'un terminal à l'autre. Par exemple, la plupart des terminaux font la distinction majuscules/majuscules, mais Powershell ne fait pas la distinction majuscules/minuscules. Cela signifie que les deux exemples de commandes suivants produiraient des résultats différents pour les terminaux distinguant les majuscules et minuscules en fonction de leur affichage **`MyFile*.txt`** et **`myfile*.txt`** en fonction de leurs paramètres.

Cependant, PowerShell traiterait ces demandes comme il le voit `MyFile*.txt` et selon les `myfile*.txt` mêmes paramètres. L'exemple de commande suivant illustre ces paramètres à l'aide de la `aws s3 cp` commande :

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "MyFile*.txt"
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "myfile*.txt"
```

Pour plus d'informations sur PowerShell l'insensibilité aux majuscules et minuscules, consultez about_Case-sensitivity dans la documentation. PowerShell

Vous devez parfois utiliser des guillemets ou des littéraux autour des chaînes contenant des caractères spéciaux ou des espaces. Les règles relatives à ce formatage peuvent également varier d'un terminal à l'autre. Pour plus d'informations sur l'utilisation de guillemets autour de paramètres complexes, consultez Utiliser des guillemets et des littéraux avec des chaînes dans AWS CLI.

Ces rubriques couvrent les règles de formatage des terminaux les plus courantes. Si vous ne parvenez pas à reconnaître les valeurs de vos paramètres par votre terminal, n'oubliez pas de consulter les rubriques de cette section et de consulter la documentation de votre terminal pour connaître les règles de syntaxe spécifiques.

Sujets relatifs aux paramètres

- Types de paramètres courants dans le AWS CLI
- Utiliser des guillemets et des littéraux avec des chaînes dans AWS CLI
- Chargement d'un paramètre depuis un fichier dans AWS CLI
- AWS CLI squelettes et fichiers d'entrée dans le AWS CLI
- En utilisant la syntaxe abrégée dans le AWS CLI

Types de paramètres courants dans le AWS CLI

Cette section décrit certains types de paramètres communs et le format typique requis.

Si vous ne parvenez pas à formater un paramètre pour une commande spécifique, consultez l'aide en saisissant le texte **`help`** après le nom de la commande. L'aide pour chaque sous-commande inclut le nom et la description d'une option. Le type de paramètre de l'option est indiqué entre parenthèses. Pour plus d'informations sur l'affichage de l'aide, consultezthe section called “Obtenir de l'aide”.

Les types de paramètres incluent :

- Chaîne
- Horodatage
- Liste
- Booléen
- Entier
- Binaire/blob (gros objet binaire) et blob de streaming
- Map
- Document

Chaîne

Les paramètres de chaîne peuvent contenir des caractères alphanumériques, des symboles et des espaces blancs issus du jeu de caractères ASCII. Les chaînes contenant des espaces blancs doivent être entourées de guillemets. Nous vous recommandons de ne pas utiliser de symboles ou d'espaces blancs autres que le caractère d'espace standard et de respecter les règles de citation de votre terminal pour éviter des résultats inattendus.

Certains paramètres de chaîne peuvent accepter les données binaires d'un fichier. Veuillez consulter Fichiers binaires pour obtenir un exemple.

Horodatage

Les horodatages sont mis en forme selon la norme ISO 8601. Ces paramètres sont souvent appelés « `DateTime` » ou `Date` « ».

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

Les formats acceptables sont les suivants :

- *YYYY*- *MM* - *DD* T *hh* : *mm* : *ss.sssTZD (UTC)*, par exemple, 2014-10-01T 20:30:00.000 Z
- *YYYY*- *MM* - *DD* T *hh* : *mm* : *ss.sssTZD (with offset)*, par exemple, 2014-10-01T 12:30:00.000-08:00
- *YYYY*- *MM* -*DD*, par exemple, 2014-10-01
- Horaire Unix en secondes, par exemple 1412195400. Ceci est parfois appelé Heure d'époque Unix et représente le nombre de secondes écoulées depuis le 1er janvier 1970 à minuit, UTC.

Par défaut, la AWS CLI version 2 traduit toutes les DateTime valeurs de réponse au format ISO 8601.

Vous pouvez définir le format d'horodatage à l'aide du paramètre du `cli_timestamp_format` fichier.

Liste

Une ou plusieurs chaînes séparées par des espaces. Si l'un des éléments de chaîne contient un espace, vous devez mettre cet élément entre guillemets. Respectez les règles de citation de votre terminal pour éviter des résultats inattendus.

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

Booléen

Drapeau binaire qui active ou désactive une option. Par exemple, `ec2 describe-spot-price-history` comporte un paramètre d'essai booléen `--dry-run` qui, lorsqu'il est spécifié, valide la demande sur le service sans réellement exécuter la requête.

```
$ aws ec2 describe-spot-price-history --dry-run
```

La sortie indique si la commande était ou non correctement mise en forme. Cette commande inclut également une version `--no-dry-run` du paramètre qui peut être utilisée pour indiquer de manière explicite que la commande doit s'exécuter normalement. Il n'est pas nécessaire de l'inclure puisqu'il s'agit du comportement par défaut.

Entier

Nombre entier non signé.

```
$ aws ec2 describe-spot-price-history --max-items 5
```

Binaire/blob (gros objet binaire) et blob de streaming

Dans le AWS CLI, vous pouvez transmettre une valeur binaire sous forme de chaîne directement sur la ligne de commande. Il existe deux types de blobs :

- Blob
- Blob en streaming

BLOB

Pour transmettre une valeur à un paramètre avec typeblob, vous devez spécifier un chemin d'accès à un fichier local contenant les données binaires à l'aide du `fileb://` préfixe. Les fichiers référencés à l'aide du `fileb://` préfixe sont toujours traités comme des fichiers binaires bruts non codés. Le chemin spécifié est interprété comme étant relatif au répertoire de travail actuel. Par exemple, le paramètre `--plaintext` pour `aws kms encrypt` est un objet blob.

```
$ aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

> **Note**
>
> Pour des raisons de rétrocompatibilité, vous pouvez utiliser le `file://` préfixe. Deux formats sont utilisés en fonction du paramètre du fichier `cli_binary_format` ou de l'option de ligne de `--cli-binary-format` commande :
>
> - Par défaut pour la AWS CLI version 2. Si la valeur du paramètre est`base64`, les fichiers référencés à l'aide du `file://` préfixe sont traités comme du texte codé en base64.
> - Par défaut pour la AWS CLI version 1. Si la valeur du paramètre est`raw-in-base64-out`, les fichiers référencés à l'aide du `file://` préfixe sont lus sous forme de texte, puis les AWS CLI tentatives de codage en binaire sont tentées.
>
> Pour plus d'informations, consultez le paramètre de fichier `cli_binary_format` ou l'option de ligne de `--cli-binary-format` commande.

Blob en streaming

Blobs de streaming tels que « `aws cloudsearchdomain upload-documents` Ne pas utiliser de préfixes ». Au lieu de cela, les paramètres du blob de streaming sont formatés à l'aide du chemin direct du fichier. L'exemple suivant utilise le chemin de fichier direct `document-batch.json` pour la `aws cloudsearchdomain upload-documents` commande :

```
$ aws cloudsearchdomain upload-documents \
```

```
    --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \
    --content-type application/json \
    --documents document-batch.json
```

Map

Ensemble de paires clés-valeurs spécifiées en JSON ou à l'aide de la syntaxe raccourcie de l'interface de ligne de commande. L'exemple JSON suivant lit un élément d'une table Amazon DynamoDB nommée my-table avec un paramètre de carte,. `--key` Le paramètre spécifie la clé primaire nommée id avec la valeur numérique 1 dans une structure JSON imbriquée.

Pour une utilisation plus avancée du JSON dans une ligne de commande, pensez à utiliser un processeur JSON en ligne de commande`jq`, par exemple pour créer des chaînes JSON. Pour plus d'informations sur`jq`, consultez le référentiel jq sur GitHub.

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N":"1"}}'

{
    "Item": {
        "name": {
            "S": "John"
        },
        "id": {
            "N": "1"
        }
    }
}
```

Document

> **Note**
>
> La syntaxe abrégée n'est pas compatible avec les types de documents.

Les types de documents sont utilisés pour envoyer des données sans qu'il soit nécessaire d'intégrer du JSON dans des chaînes. Le type de document permet aux services de fournir des schémas arbitraires afin que vous puissiez utiliser des types de données plus flexibles.

Cela permet d'envoyer des données JSON sans avoir à échapper aux valeurs. Par exemple, au lieu d'utiliser l'entrée JSON échappée suivante :

```
{"document": "{\"key\":true}"}
```

Vous pouvez utiliser le type de document suivant :

```
{"document": {"key": true}}
```

Valeurs valides pour les types de documents

En raison de la flexibilité des types de documents, il existe plusieurs types de valeurs valides. Les valeurs valides sont notamment les suivantes :

Chaîne

```
--option '"value"'
```

Nombre

```
--option 123
--option 123.456
```

Booléen

```
--option true
```

Null

```
--option null
```

Tableau

```
--option '["value1", "value2", "value3"]'
--option '["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]'
```

Objet

```
--option '{"key": "value"}'
--option '{"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":
 ["value3", "value4"], "key6": {"value5": "value6"}'
```

Utiliser des guillemets et des littéraux avec des chaînes dans AWS CLI

Les guillemets simples et doubles sont principalement utilisés de deux manières dans le AWS CLI.

- Utilisation de guillemets autour des chaînes contenant des espaces blancs
- Utilisation de guillemets dans les chaînes

Utilisation de guillemets autour des chaînes contenant des espaces blancs

Les noms des paramètres et leurs valeurs sont séparés par des espaces sur la ligne de commande. Si une valeur de chaîne contient un espace intégré, vous devez entourer la chaîne entière de guillemets pour éviter que l' AWS CLI espace ne soit mal interprété comme un séparateur entre la valeur et le nom du paramètre suivant. Le type de guillemet que vous utilisez dépend du système d'exploitation AWS CLI sur lequel vous l'exécutez.

Linux and macOS

Utiliser des guillemets simples `' '`

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

Pour plus d'informations sur l'utilisation des guillemets, consultez la documentation utilisateur de votre shell préféré.

PowerShell

Devis uniques (recommandé)

`' '`Les guillemets simples sont appelés `verbatim` chaînes. La chaîne est transmise à la commande exactement telle que vous la tapez, ce qui signifie que PowerShell les variables ne seront pas transmises.

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

Devis doubles

`" "`Les guillemets doubles sont appelés `expandable` chaînes de caractères. Les variables peuvent être transmises sous forme de chaînes extensibles.

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Pour plus d'informations sur l'utilisation des guillemets, voir À propos des règles de citation dans Microsoft PowerShell Docs.

Windows command prompt

Utilisez des guillemets doubles" ".

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Vous pouvez éventuellement séparer le nom du paramètre de la valeur par un signe égal = au lieu d'un espace. Cela n'est généralement nécessaire que si la valeur du paramètre commence par un trait d'union.

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

Utilisation de guillemets dans les chaînes

Les chaînes peuvent contenir des guillemets, et votre interpréteur de commandes peut avoir besoin de guillemets pour fonctionner correctement. L'un des types de valeurs de paramètre courants est une chaîne JSON. Ceci est complexe car il inclut des espaces et des guillemets doubles " " autour du nom et de la valeur de chaque élément dans la structure JSON. La manière dont vous entrez des paramètres au format JSON sur la ligne de commande varie selon le système d'exploitation.

Pour une utilisation plus avancée du JSON dans la ligne de commande, pensez à utiliser un processeur JSON en ligne de commande`jq`, par exemple pour créer des chaînes JSON. Pour plus d'informations sur`jq`, consultez le référentiel jq sur GitHub.

Linux and macOS

Pour que Linux et macOS interprètent littéralement les chaînes, utilisez des guillemets simples ' ' pour entourer la structure de données JSON, comme dans l'exemple suivant. Il n'est pas nécessaire d'échapper aux guillemets doubles intégrés dans la chaîne JSON, car ils sont traités littéralement. Comme le JSON est placé entre guillemets simples, tous les guillemets simples de la chaîne devront être évités. Cela se fait généralement en utilisant une barre oblique inverse avant le guillemet simple. `\'`

```
$ aws ec2 run-instances \
```

```
    --image-id ami-12345678 \
    --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}}]'
```

Pour plus d'informations sur l'utilisation des guillemets, consultez la documentation utilisateur de votre shell préféré.

PowerShell

Utilisez des guillemets simples ' ' ou des guillemets doubles" ".

Devis uniques (recommandé)

' 'Les guillemets simples sont appelés `verbatim` chaînes. La chaîne est transmise à la commande exactement telle que vous la tapez, ce qui signifie que PowerShell les variables ne seront pas transmises.

Comme les structures de données JSON incluent des guillemets doubles, nous suggérons de les ' ' placer entre guillemets simples. Si vous utilisez des guillemets simples, vous n'avez pas besoin d'échapper aux guillemets doubles intégrés dans la chaîne JSON. Cependant, vous devez éviter chaque guillemet par un backtick ` dans la structure JSON.

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}}]'
```

Devis doubles

" "Les guillemets sont appelés `expandable` chaînes. Les variables peuvent être transmises sous forme de chaînes extensibles.

Si vous utilisez des guillemets doubles, il n'est pas nécessaire d'échapper aux guillemets simples intégrés dans la chaîne JSON. Cependant, vous devez éviter chaque guillemet double par un backtick ` dans la structure JSON, comme dans l'exemple suivant.

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings "[{`"DeviceName`":`"/dev/sdb`",`"Ebs`":
{`"VolumeSize`":20,`"DeleteOnTermination`":false,`"VolumeType`":`"standard`"}}]"
```

Pour plus d'informations sur l'utilisation des guillemets, voir À propos des règles de citation dans Microsoft PowerShell Docs.

> **Warning**
>
> Avant d' PowerShell envoyer une commande au AWS CLI, il détermine si votre commande est interprétée à l'aide de règles classiques PowerShell ou entre `CommandLineToArgvW` guillemets. Lorsque PowerShell des processus l'utilisent`CommandLineToArgvW`, vous devez éviter les caractères marqués d'une barre oblique inverse\.
>
> Pour plus d'informations sur *`CommandLineToArgvW`* in PowerShell, consultez What's up with the strange treatment of guillemets and backslashes par CommandLineToArgv W dans Microsoft DevBlogs, Everyone cite les arguments de ligne de commande de la mauvaise façon dans le blog Microsoft Docs, et sur la fonction CommandLineToArgv W dans Microsoft Docs.
>
> Devis uniques
>
> ' 'Les guillemets simples sont appelés `verbatim` chaînes. La chaîne est transmise à la commande exactement telle que vous la tapez, ce qui signifie que PowerShell les variables ne seront pas transmises. Échappez aux personnages avec une barre oblique inverse\.
>
> ```
> PS C:\> aws ec2 run-instances `
>     --image-id ami-12345678 `
>     --block-device-mappings '[{\"DeviceName\":\"/dev/sdb\",\"Ebs\":
> {\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":\"standard\"}}]'
> ```
>
> Devis doubles
>
> " "Les guillemets sont appelés `expandable` chaînes. Les variables peuvent être transmises sous forme de `expandable` chaînes. Pour les chaînes entre guillemets, vous devez éviter d'utiliser deux fois `` `\ `` pour chaque guillemet au lieu de simplement utiliser un backtick. Le backtick échappe à la barre oblique inverse, puis la barre oblique inverse est utilisée comme caractère d'échappement pour le processus. `CommandLineToArgvW`
>
> ```
> PS C:\> aws ec2 run-instances `
>     --image-id ami-12345678 `
>     --block-device-mappings "[{`\"DeviceName`\":`\"/dev/sdb`\",`\"Ebs`\":
> {`\"VolumeSize`\":20,`\"DeleteOnTermination`\":false,`\"VolumeType`\":`
> \"standard`\"}}]"
> ```

Blobs (recommandé)
Pour contourner les règles PowerShell de citation pour la saisie de données JSON, utilisez Blobs pour transmettre vos données JSON directement au. AWS CLI Pour plus d'informations sur les blobs, consultezBLOB.

Windows command prompt

L'invite de commande Windows nécessite des guillemets doubles " " pour entourer la structure de données JSON. En outre, pour éviter que le processeur de commandes n'interprète mal les guillemets doubles intégrés au JSON, vous devez également éviter (faire précéder d'une barre \ oblique inversée) chaque guillemet double " dans la structure de données JSON elle-même, comme dans l'exemple suivant.

```
C:\> aws ec2 run-instances ^
    --image-id ami-12345678 ^
    --block-device-mappings "[{\"DeviceName\":\"/dev/sdb\",\"Ebs\":
{\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":\"standard\"}}]"
```

Seuls les guillemets doubles les plus à l'extérieur ne sont pas placés dans une séquence d'échappement.

Chargement d'un paramètre depuis un fichier dans AWS CLI

Certains paramètres attendent des noms de fichiers comme arguments, à partir desquels ils AWS CLI chargent les données. D'autres paramètres vous permettent de spécifier la valeur du paramètre sous forme de texte saisi sur la ligne de commande ou de lecture à partir d'un fichier. Qu'un fichier soit obligatoire ou facultatif, vous devez le coder correctement pour qu'ils AWS CLI puissent le comprendre. L'encodage du fichier doit correspondre aux paramètres régionaux par défaut du système de lecture. Vous pouvez le déterminer à l'aide de la `locale.getpreferredencoding()` méthode Python.

Cette méthode permet de charger un fichier pour un seul paramètre. Pour plus d'informations sur le chargement de plusieurs paramètres dans un seul fichier, consultezthe section called “Générer un modèle de squelette de l'interface de ligne de commande”.

> **Note**
>
> Par défaut, Windows PowerShell affiche le texte au format UTF-16, ce qui est en conflit avec le codage UTF-8 utilisé par les fichiers JSON et de nombreux systèmes Linux. Nous vous recommandons de l'utiliser `-Encoding ascii` avec vos PowerShell `Out-File` commandes pour vous assurer qu' AWS CLI ils peuvent lire le fichier obtenu.

Rubriques

- Comment charger un paramètre depuis un fichier
- Fichiers binaires
- Chargement d'un fichier en tant que valeur de syntaxe abrégée

Comment charger un paramètre depuis un fichier

Il est parfois pratique de charger une valeur de paramètre à partir d'un fichier plutôt que d'essayer de la taper entièrement comme valeur de paramètre de ligne de commande, par exemple lorsque le paramètre est une chaîne JSON complexe. Pour spécifier un fichier qui contient la valeur, spécifiez une URL de fichier au format suivant :

```
file://complete/path/to/file
```

- Les deux premières barres obliques « / » font partie de la spécification. Si le chemin d'accès requis commence par un caractère « / », le résultat contiendra trois barres obliques : `file:///folder/file`.
- L'URL fournit le chemin d'accès au fichier qui contient le contenu effectif du paramètre.
- Lorsque vous utilisez des fichiers contenant des espaces ou des caractères spéciaux, suivez les règles de citation et d'échappement de votre terminal.

Les chemins de fichiers des exemples suivants sont interprétés comme étant relatifs au répertoire de travail actuel.

Linux or macOS

```
// Read from a file in the current directory
$ aws ec2 describe-instances --filters file://filter.json
```

```
// Read from a file in /tmp
$ aws ec2 describe-instances --filters file:///tmp/filter.json

// Read from a file with a filename with whitespaces
$ aws ec2 describe-instances --filters 'file://filter content.json'
```

Windows command prompt

```
// Read from a file in C:\temp
C:\> aws ec2 describe-instances --filters file://C:\temp\filter.json

// Read from a file with a filename with whitespaces
C:\> aws ec2 describe-instances --filters "file://C:\temp\filter content.json"
```

L'option de préfixe `file://` prend en charge les extensions de style Unix, dont « `~/` », « `./` » et « `../` ». Sous Windows, l'expression « `~/` » s'étend à votre répertoire utilisateur, stocké dans la variable d'environnement `%USERPROFILE%`. Par exemple, sous Windows 10, un répertoire utilisateur figure généralement sous `%USERPROFILE%`.

Les documents JSON qui sont intégrés en tant que valeur d'un autre document JSON doivent toujours inclure une séquence d'échappement.

```
$ aws sqs create-queue --queue-name my-queue --attributes file://attributes.json
```

attributes.json

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
west-2:0123456789012:deadletter\", \"maxReceiveCount\":\"5\"}"
}
```

Fichiers binaires

Pour les commandes qui acceptent des données binaires comme paramètre, spécifiez que les données sont un contenu binaire à l'aide du préfixe `fileb://`. Les commandes qui acceptent des données binaires sont les suivantes :

- **`aws ec2 run-instances`:**`--user-data`paramètre.
- **`aws s3api put-object`:**`--sse-customer-key`paramètre.

- **aws kms decrypt:**`--ciphertext-blob`paramètre.

L'exemple suivant génère une clé AES binaire 256 bits à l'aide d'un outil de ligne de commande Linux, puis la fournit à Amazon S3 pour chiffrer un fichier chargé côté serveur.

```
$ dd if=/dev/urandom bs=1 count=32 > sse.key
32+0 records in
32+0 records out
32 bytes (32 B) copied, 0.000164441 s, 195 kB/s
$ aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key test.txt \
    --body test.txt \
    --sse-customer-key fileb://sse.key \
    --sse-customer-algorithm AES256
{
    "SSECustomerKeyMD5": "iVg8oWa8sy714+FjtesrJg==",
    "SSECustomerAlgorithm": "AES256",
    "ETag": "\"a6118e84b76cf98bf04bbe14b6045c6c\""
}
```

Pour un autre exemple de référence à un fichier contenant des paramètres au format JSON, consultez. Associer une politique gérée par IAM à un utilisateur

Chargement d'un fichier en tant que valeur de syntaxe abrégée

Lorsque vous utilisez une syntaxe abrégée lorsqu'une valeur est grande ou complexe, il est souvent plus facile de la charger dans un fichier en tant que valeur. Pour charger un fichier en tant que valeur de syntaxe abrégée, le formatage changera légèrement. Au lieu de cela, `key=value` vous utiliserez l'@=opérateur au lieu de l'=opérateur. Cela @= signifie AWS CLI que la valeur doit être lue comme un chemin de fichier et non comme une chaîne. L'exemple suivant montre une paire clé-valeur chargeant un fichier pour sa valeur.

Linux or macOS

```
--option key@=file://template.txt
```

Windows

```
--option "key1@=file://template.txt"
```

L'exemple suivant montre comment charger un fichier de certificat pour la `aws rolesanywhere create-trust-anchor` commande.

```
$ aws rolesanywhere create-trust-anchor --name TrustAnchor \
    --source sourceData={x509CertificateData@=file://root-ca.crt},sourceType="CERTIFICATE_BUNDLE"  \
    --enabled
```

Pour plus d'informations sur la syntaxe abrégée, consultez. the section called “Syntaxe raccourcie”

AWS CLI squelettes et fichiers d'entrée dans le AWS CLI

La plupart des AWS CLI commandes acceptent l'importation de paramètres depuis un fichier. Ces modèles peuvent être générés à l'aide de l'`generate-cli-skeleton`option, puis importés à l'aide `--cli-input-json` des `--cli-input-yaml` paramètres et.

Rubriques

- À propos des AWS CLI squelettes et des fichiers d'entrée
- Génération et importation d'un squelette de commande
- Combinaison de fichiers d'entrée et de paramètres de ligne de commande

À propos des AWS CLI squelettes et des fichiers d'entrée

La plupart des commandes AWS Command Line Interface (AWS CLI) permettent d'accepter les entrées de paramètres d'un fichier à l'aide du `--cli-input-yaml` paramètre s `--cli-input-json` et.

Ces mêmes commandes utilisent le `--generate-cli-skeleton` paramètre pour générer un fichier au format JSON ou YAML avec tous les paramètres que vous pouvez modifier et renseigner. Vous pouvez ensuite exécuter la commande avec le `--cli-input-yaml` paramètre `--cli-input-json` ou et pointer sur le fichier rempli.

> **Important**
>
> Les AWS CLI commandes personnalisées, telles que les , ne prennent en charge ni les paramètres `--generate-cli-skeleton` or ni `--cli-input-json` les `--cli-input-yaml` paramètres décrits dans cette rubrique. Pour vérifier si une commande spécifique

prend en charge ces paramètres, exécutez la `help`commande que vous souhaitez utiliser ou reportez-vous au version 2 du guide de référence.

`--generate-cli-skeleton`Génère et affiche un modèle de paramètres que vous pouvez personnaliser et utiliser comme entrée dans une commande. Le modèle généré inclut tous les paramètres pris en charge par la commande.

Le paramètre `--generate-cli-skeleton` accepte l'une des valeurs suivantes :

- `input`— Le modèle généré inclut tous les paramètres d'entrée au format JSON. C'est la valeur par défaut.
- `yaml-input`— Le modèle généré inclut tous les paramètres d'entrée au format YAML.
- `output`— Le modèle généré inclut tous les paramètres de sortie au format JSON. Vous ne pouvez actuellement pas demander les paramètres de sortie au format YAML.

Comme il AWS CLI s'agit essentiellement d'une « enveloppe » autour de l'API du service, le fichier squelette s'attend à ce que vous référenciez tous les paramètres par leurs noms de paramètres d'API sous-jacents. Il est probablement différent du nom du AWS CLI paramètre. Par exemple, un AWS CLI paramètre nommé `user-name` peut être mappé au paramètre d'API du AWS service nommé `UserName` (notez la modification des majuscules et l'absence du tiret). Nous vous recommandons d'utiliser l'option `--generate-cli-skeleton` pour générer le modèle avec les noms de paramètres « corrects » afin d'éviter des erreurs. Vous pouvez consulter le guide de référence des API du service pour connaître les noms de paramètres attendus. Vous pouvez supprimer dans le modèle tous les paramètres qui ne sont pas obligatoires et pour lesquels vous ne souhaitez pas fournir de valeur.

Par exemple, si vous exécutez la commande suivante, elle génère le modèle de paramètres pour la commande Amazon Elastic Compute Cloud (Amazon EC2)run-instances.

JSON

L'exemple suivant montre comment générer un modèle formaté en JSON à l'aide de la valeur par défaut (`input`) pour le paramètre `--generate-cli-skeleton`.

```
$ aws ec2 run-instances --generate-cli-skeleton
```

```
{
```

```
    "DryRun": true,
    "ImageId": "",
    "MinCount": 0,
    "MaxCount": 0,
    "KeyName": "",
    "SecurityGroups": [
        ""
    ],
    "SecurityGroupIds": [
        ""
    ],
    "UserData": "",
    "InstanceType": "",
    "Placement": {
        "AvailabilityZone": "",
        "GroupName": "",
        "Tenancy": ""
    },
    "KernelId": "",
    "RamdiskId": "",
    "BlockDeviceMappings": [
        {
            "VirtualName": "",
            "DeviceName": "",
            "Ebs": {
                "SnapshotId": "",
                "VolumeSize": 0,
                "DeleteOnTermination": true,
                "VolumeType": "",
                "Iops": 0,
                "Encrypted": true
            },
            "NoDevice": ""
        }
    ],
    "Monitoring": {
        "Enabled": true
    },
    "SubnetId": "",
    "DisableApiTermination": true,
    "InstanceInitiatedShutdownBehavior": "",
    "PrivateIpAddress": "",
    "ClientToken": "",
    "AdditionalInfo": "",
```

```
    "NetworkInterfaces": [
        {
            "NetworkInterfaceId": "",
            "DeviceIndex": 0,
            "SubnetId": "",
            "Description": "",
            "PrivateIpAddress": "",
            "Groups": [
                ""
            ],
            "DeleteOnTermination": true,
            "PrivateIpAddresses": [
                {
                    "PrivateIpAddress": "",
                    "Primary": true
                }
            ],
            "SecondaryPrivateIpAddressCount": 0,
            "AssociatePublicIpAddress": true
        }
    ],
    "IamInstanceProfile": {
        "Arn": "",
        "Name": ""
    },
    "EbsOptimized": true
}
```

YAML

L'exemple suivant montre comment générer un modèle formaté en YAML à l'aide de la valeur `yaml-input` du paramètre `--generate-cli-skeleton`.

```
$ aws ec2 run-instances --generate-cli-skeleton yaml-input
```

```
BlockDeviceMappings:  # The block device mapping entries.
- DeviceName: ''  # The device name (for example, /dev/sdh or xvdh).
  VirtualName: '' # The virtual device name (ephemeralN).
  Ebs: # Parameters used to automatically set up Amazon EBS volumes when the
 instance is launched.
    DeleteOnTermination: true  # Indicates whether the EBS volume is deleted on
 instance termination.
```

```
      Iops: 0 # The number of I/O operations per second (IOPS) that the volume
    supports.
      SnapshotId: '' # The ID of the snapshot.
      VolumeSize: 0 # The size of the volume, in GiB.
      VolumeType: st1 # The volume type. Valid values are: standard, io1, gp2, sc1,
    st1.
      Encrypted: true # Indicates whether the encryption state of an EBS volume is
    changed while being restored from a backing snapshot.
      KmsKeyId: '' # Identifier (key ID, key alias, ID ARN, or alias ARN) for a
    customer managed KMS key under which the EBS volume is encrypted.
     NoDevice: '' # Suppresses the specified device included in the block device
    mapping of the AMI.
ImageId: '' # The ID of the AMI.
InstanceType: c4.4xlarge # The instance type. Valid values are: t1.micro, t2.nano,
    t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge, t3.nano, t3.micro,
    t3.small, t3.medium, t3.large, t3.xlarge, t3.2xlarge, t3a.nano, t3a.micro,
    t3a.small, t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge, m1.small, m1.medium,
    m1.large, m1.xlarge, m3.medium, m3.large, m3.xlarge, m3.2xlarge, m4.large,
    m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge, m4.16xlarge, m2.xlarge, m2.2xlarge,
    m2.4xlarge, cr1.8xlarge, r3.large, r3.xlarge, r3.2xlarge, r3.4xlarge, r3.8xlarge,
    r4.large, r4.xlarge, r4.2xlarge, r4.4xlarge, r4.8xlarge, r4.16xlarge, r5.large,
    r5.xlarge, r5.2xlarge, r5.4xlarge, r5.8xlarge, r5.12xlarge, r5.16xlarge,
    r5.24xlarge, r5.metal, r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge,
    r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge, r5d.large, r5d.xlarge,
    r5d.2xlarge, r5d.4xlarge, r5d.8xlarge, r5d.12xlarge, r5d.16xlarge, r5d.24xlarge,
    r5d.metal, r5ad.large, r5ad.xlarge, r5ad.2xlarge, r5ad.4xlarge, r5ad.8xlarge,
    r5ad.12xlarge, r5ad.16xlarge, r5ad.24xlarge, x1.16xlarge, x1.32xlarge, x1e.xlarge,
    x1e.2xlarge, x1e.4xlarge, x1e.8xlarge, x1e.16xlarge, x1e.32xlarge, i2.xlarge,
    i2.2xlarge, i2.4xlarge, i2.8xlarge, i3.large, i3.xlarge, i3.2xlarge, i3.4xlarge,
    i3.8xlarge, i3.16xlarge, i3.metal, i3en.large, i3en.xlarge, i3en.2xlarge,
    i3en.3xlarge, i3en.6xlarge, i3en.12xlarge, i3en.24xlarge, i3en.metal, hi1.4xlarge,
    hs1.8xlarge, c1.medium, c1.xlarge, c3.large, c3.xlarge, c3.2xlarge, c3.4xlarge,
    c3.8xlarge, c4.large, c4.xlarge, c4.2xlarge, c4.4xlarge, c4.8xlarge, c5.large,
    c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge, c5.12xlarge, c5.18xlarge,
    c5.24xlarge, c5.metal, c5d.large, c5d.xlarge, c5d.2xlarge, c5d.4xlarge,
    c5d.9xlarge, c5d.18xlarge, c5n.large, c5n.xlarge, c5n.2xlarge, c5n.4xlarge,
    c5n.9xlarge, c5n.18xlarge, cc1.4xlarge, cc2.8xlarge, g2.2xlarge, g2.8xlarge,
    g3.4xlarge, g3.8xlarge, g3.16xlarge, g3s.xlarge, g4dn.xlarge, g4dn.2xlarge,
    g4dn.4xlarge, g4dn.8xlarge, g4dn.12xlarge, g4dn.16xlarge, cg1.4xlarge, p2.xlarge,
    p2.8xlarge, p2.16xlarge, p3.2xlarge, p3.8xlarge, p3.16xlarge, p3dn.24xlarge,
    d2.xlarge, d2.2xlarge, d2.4xlarge, d2.8xlarge, f1.2xlarge, f1.4xlarge, f1.16xlarge,
    m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5.8xlarge, m5.12xlarge, m5.16xlarge,
    m5.24xlarge, m5.metal, m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge,
    m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge, m5d.large, m5d.xlarge,
```

```
 m5d.2xlarge, m5d.4xlarge, m5d.8xlarge, m5d.12xlarge, m5d.16xlarge, m5d.24xlarge,
 m5d.metal, m5ad.large, m5ad.xlarge, m5ad.2xlarge, m5ad.4xlarge, m5ad.8xlarge,
 m5ad.12xlarge, m5ad.16xlarge, m5ad.24xlarge, h1.2xlarge, h1.4xlarge, h1.8xlarge,
 h1.16xlarge, z1d.large, z1d.xlarge, z1d.2xlarge, z1d.3xlarge, z1d.6xlarge,
 z1d.12xlarge, z1d.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal,
 u-24tb1.metal, a1.medium, a1.large, a1.xlarge, a1.2xlarge, a1.4xlarge, a1.metal,
 m5dn.large, m5dn.xlarge, m5dn.2xlarge, m5dn.4xlarge, m5dn.8xlarge, m5dn.12xlarge,
 m5dn.16xlarge, m5dn.24xlarge, m5n.large, m5n.xlarge, m5n.2xlarge, m5n.4xlarge,
 m5n.8xlarge, m5n.12xlarge, m5n.16xlarge, m5n.24xlarge, r5dn.large, r5dn.xlarge,
 r5dn.2xlarge, r5dn.4xlarge, r5dn.8xlarge, r5dn.12xlarge, r5dn.16xlarge,
 r5dn.24xlarge, r5n.large, r5n.xlarge, r5n.2xlarge, r5n.4xlarge, r5n.8xlarge,
 r5n.12xlarge, r5n.16xlarge, r5n.24xlarge.
Ipv6AddressCount: 0 # [EC2-VPC] The number of IPv6 addresses to associate with the
 primary network interface.
Ipv6Addresses: # [EC2-VPC] The IPv6 addresses from the range of the subnet to
 associate with the primary network interface.
- Ipv6Address: ''  # The IPv6 address.
KernelId: '' # The ID of the kernel.
KeyName: '' # The name of the key pair.
MaxCount: 0 # [REQUIRED] The maximum number of instances to launch.
MinCount: 0 # [REQUIRED] The minimum number of instances to launch.
Monitoring: # Specifies whether detailed monitoring is enabled for the instance.
  Enabled: true  # [REQUIRED] Indicates whether detailed monitoring is enabled.
Placement: # The placement for the instance.
  AvailabilityZone: ''  # The Availability Zone of the instance.
  Affinity: '' # The affinity setting for the instance on the Dedicated Host.
  GroupName: '' # The name of the placement group the instance is in.
  PartitionNumber: 0 # The number of the partition the instance is in.
  HostId: '' # The ID of the Dedicated Host on which the instance resides.
  Tenancy: dedicated # The tenancy of the instance (if the instance is running in a
 VPC). Valid values are: default, dedicated, host.
  SpreadDomain: '' # Reserved for future use.
RamdiskId: '' # The ID of the RAM disk to select.
SecurityGroupIds: # The IDs of the security groups.
- ''
SecurityGroups: # [default VPC] The names of the security groups.
- ''
SubnetId: '' # [EC2-VPC] The ID of the subnet to launch the instance into.
UserData: '' # The user data to make available to the instance.
AdditionalInfo: '' # Reserved.
ClientToken: '' # Unique, case-sensitive identifier you provide to ensure the
 idempotency of the request.
DisableApiTermination: true # If you set this parameter to true, you can't terminate
 the instance using the Amazon EC2 console, CLI, or API; otherwise, you can.
```

```
DryRun: true # Checks whether you have the required permissions for the action,
 without actually making the request, and provides an error response.
EbsOptimized: true # Indicates whether the instance is optimized for Amazon EBS I/O.
IamInstanceProfile: # The IAM instance profile.
  Arn: ''  # The Amazon Resource Name (ARN) of the instance profile.
  Name: '' # The name of the instance profile.
InstanceInitiatedShutdownBehavior: stop # Indicates whether an instance stops or
 terminates when you initiate shutdown from the instance (using the operating system
 command for system shutdown). Valid values are: stop, terminate.
NetworkInterfaces: # The network interfaces to associate with the instance.
- AssociatePublicIpAddress: true  # Indicates whether to assign a public IPv4
 address to an instance you launch in a VPC.
  DeleteOnTermination: true # If set to true, the interface is deleted when the
 instance is terminated.
  Description: '' # The description of the network interface.
  DeviceIndex: 0 # The position of the network interface in the attachment order.
  Groups: # The IDs of the security groups for the network interface.
  - ''
  Ipv6AddressCount: 0 # A number of IPv6 addresses to assign to the network
 interface.
  Ipv6Addresses: # One or more IPv6 addresses to assign to the network interface.
  - Ipv6Address: ''  # The IPv6 address.
  NetworkInterfaceId: '' # The ID of the network interface.
  PrivateIpAddress: '' # The private IPv4 address of the network interface.
  PrivateIpAddresses: # One or more private IPv4 addresses to assign to the network
 interface.
  - Primary: true  # Indicates whether the private IPv4 address is the primary
 private IPv4 address.
    PrivateIpAddress: '' # The private IPv4 addresses.
  SecondaryPrivateIpAddressCount: 0 # The number of secondary private IPv4
 addresses.
  SubnetId: '' # The ID of the subnet associated with the network interface.
  InterfaceType: '' # The type of network interface.
PrivateIpAddress: '' # [EC2-VPC] The primary IPv4 address.
ElasticGpuSpecification: # An elastic GPU to associate with the instance.
- Type: ''  # [REQUIRED] The type of Elastic Graphics accelerator.
ElasticInferenceAccelerators: # An elastic inference accelerator to associate with
 the instance.
- Type: ''  # [REQUIRED]  The type of elastic inference accelerator.
TagSpecifications: # The tags to apply to the resources during launch.
- ResourceType: network-interface  # The type of resource to tag. Valid values
 are: client-vpn-endpoint, customer-gateway, dedicated-host, dhcp-options, elastic-
ip, fleet, fpga-image, host-reservation, image, instance, internet-gateway,
 launch-template, natgateway, network-acl, network-interface, reserved-instances,
```

```
 route-table, security-group, snapshot, spot-instances-request, subnet, traffic-
mirror-filter, traffic-mirror-session, traffic-mirror-target, transit-gateway,
 transit-gateway-attachment, transit-gateway-route-table, volume, vpc, vpc-peering-
connection, vpn-connection, vpn-gateway.
  Tags: # The tags to apply to the resource.
  - Key: ''  # The key of the tag.
    Value: '' # The value of the tag.
LaunchTemplate: # The launch template to use to launch the instances.
  LaunchTemplateId: ''  # The ID of the launch template.
  LaunchTemplateName: '' # The name of the launch template.
  Version: '' # The version number of the launch template.
InstanceMarketOptions: # The market (purchasing) option for the instances.
  MarketType: spot  # The market type. Valid values are: spot.
  SpotOptions: # The options for Spot Instances.
    MaxPrice: ''  # The maximum hourly price you're willing to pay for the Spot
 Instances.
    SpotInstanceType: one-time # The Spot Instance request type. Valid values are:
 one-time, persistent.
    BlockDurationMinutes: 0 # The required duration for the Spot Instances (also
 known as Spot blocks), in minutes.
    ValidUntil: 1970-01-01 00:00:00 # The end date of the request.
    InstanceInterruptionBehavior: terminate # The behavior when a Spot Instance is
 interrupted. Valid values are: hibernate, stop, terminate.
CreditSpecification: # The credit option for CPU usage of the T2 or T3 instance.
  CpuCredits: ''  # [REQUIRED] The credit option for CPU usage of a T2 or T3
 instance.
CpuOptions: # The CPU options for the instance.
  CoreCount: 0  # The number of CPU cores for the instance.
  ThreadsPerCore: 0 # The number of threads per CPU core.
CapacityReservationSpecification: # Information about the Capacity Reservation
 targeting option.
  CapacityReservationPreference: none  # Indicates the instance's Capacity
 Reservation preferences. Valid values are: open, none.
  CapacityReservationTarget: # Information about the target Capacity Reservation.
    CapacityReservationId: ''  # The ID of the Capacity Reservation.
HibernationOptions: # Indicates whether an instance is enabled for hibernation.
  Configured: true  # If you set this parameter to true, your instance is enabled
 for hibernation.
LicenseSpecifications: # The license configurations.
- LicenseConfigurationArn: ''  # The Amazon Resource Name (ARN) of the license
 configuration.
```

Génération et importation d'un squelette de commande

Pour générer et utiliser un fichier de squelette de paramètre

1. Exécutez la commande avec le `--generate-cli-skeleton` paramètre pour produire du JSON ou du YAML et dirigez la sortie vers un fichier pour l'enregistrer.

   JSON

   ```
   $ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
   ```

   YAML

   ```
   $ aws ec2 run-instances --generate-cli-skeleton yaml-input > ec2runinst.yaml
   ```

2. Ouvrez le fichier de squelette de paramètre dans votre éditeur de texte et supprimez les paramètres dont vous n'avez pas besoin. Par exemple, vous pouvez supprimer le modèle comme suit. Vérifiez que le fichier est toujours valide en JSON ou YAML après avoir supprimé les éléments dont vous n'avez pas besoin.

   JSON

   ```
   {
       "DryRun": true,
       "ImageId": "",
       "KeyName": "",
       "SecurityGroups": [
           ""
       ],
       "InstanceType": "",
       "Monitoring": {
           "Enabled": true
       }
   }
   ```

   YAML

   ```
   DryRun: true
   ImageId: ''
   KeyName: ''
   SecurityGroups:
   ```

```
- ''
InstanceType:
Monitoring:
  Enabled: true
```

Dans cet exemple, nous laissons le `DryRun` paramètre défini sur `true` pour utiliser la fonctionnalité Amazon EC2 Dry Run. Cette fonctionnalité vous permet de tester la commande en toute sécurité sans créer ou modifier de ressources.

3. Remplissez les valeurs restantes avec les valeurs appropriées pour votre scénario. Dans cet exemple, nous fournissons le type d'instance, le nom de la clé, le groupe de sécurité et l'identificateur de l'AMI à utiliser. Cet exemple suppose la valeur par défaut Région AWS. L'AMI `ami-dfc39aef` est une image Amazon Linux 64 bits hébergée dans la `us-west-2` région. Si vous utilisez une autre région, vous devez identifier le bon ID d'AMI à utiliser.

   JSON

   ```
   {
       "DryRun": true,
       "ImageId": "ami-dfc39aef",
       "KeyName": "mykey",
       "SecurityGroups": [
           "my-sg"
       ],
       "InstanceType": "t2.micro",
       "Monitoring": {
           "Enabled": true
       }
   }
   ```

   YAML

   ```
   DryRun: true
   ImageId: 'ami-dfc39aef'
   KeyName: 'mykey'
   SecurityGroups:
   - 'my-sg'
   InstanceType: 't2.micro'
   Monitoring:
     Enabled: true
   ```

4. Exécutez la commande avec les paramètres complétés en transmettant le fichier modèle terminé au `cli-input-yaml` paramètre `--cli-input-json` or -- en utilisant le `file://` préfixe. Le chemin AWS CLI est interprété comme étant relatif à votre répertoire de travail actuel. L'exemple suivant AWS CLI recherche le fichier dans le répertoire de travail actuel.

   JSON

   ```
   $ aws ec2 run-instances --cli-input-json file://ec2runinst.json
   ```

   ```
   A client error (DryRunOperation) occurred when calling the RunInstances
    operation: Request would have succeeded, but DryRun flag is set.
   ```

   YAML

   ```
   $ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml
   ```

   ```
   A client error (DryRunOperation) occurred when calling the RunInstances
    operation: Request would have succeeded, but DryRun flag is set.
   ```

   L'erreur d'exécution à sec indique que le JSON ou le YAML est correctement formé et que les valeurs des paramètres sont valides. Si d'autres problèmes sont signalés dans la sortie, corrigez-les et répétez l'étape précédente jusqu'à ce que le message `Request would have succeeded` s'affiche.

5. Vous pouvez maintenant définir le paramètre `DryRun` comme `false` pour désactiver l'essai.

   JSON

   ```
   {
       "DryRun": false,
       "ImageId": "ami-dfc39aef",
       "KeyName": "mykey",
       "SecurityGroups": [
           "my-sg"
       ],
       "InstanceType": "t2.micro",
       "Monitoring": {
           "Enabled": true
       }
   ```

```
}
```

YAML

```
DryRun: false
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
- 'my-sg'
InstanceType: 't2.micro'
Monitoring:
  Enabled: true
```

6. Exécutez la commande, lancez `run-instances` réellement une EC2 instance Amazon et affichez les détails générés par le lancement réussi. Le format de la sortie est contrôlé par le paramètre `--output`, séparément du format de votre modèle de paramètre d'entrée.

   JSON

   ```
   $ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
   ```

   ```
   {
       "OwnerId": "123456789012",
       "ReservationId": "r-d94a2b1",
       "Groups": [],
       "Instances": [
   ...
   ```

   YAML

   ```
   $ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --output yaml
   ```

   ```
   OwnerId: '123456789012'
   ReservationId: 'r-d94a2b1',
   Groups":
   - ''
   Instances:
   ...
   ```

Combinaison de fichiers d'entrée et de paramètres de ligne de commande

Un fichier d'entrée peut être utilisé pour tous les paramètres ou peut être combiné avec les paramètres spécifiés dans le AWS CLI. Vous pouvez utiliser cette fonctionnalité pour les paramètres que vous réutilisez fréquemment dans un fichier d'entrée, tout en conservant vos paramètres individuels dans la commande elle-même.

Les `aws ec2 run-instances` exemples suivants combinent l'utilisation d'un fichier d'entrée et de paramètres. Nous indiquons le type d'instance, le nom de clé, le groupe de sécurité, l'identifiant de l'Amazon Machine Image (AMI) à utiliser et nous supposons qu'il s'agit de l'image par défaut Région AWS. L'AMI `ami-dfc39aef` est une image Amazon Linux 64 bits hébergée dans la `us-west-2` région. Si vous utilisez une autre région, vous devez identifier le bon ID d'AMI à utiliser.

JSON

Contenu du fichier JSON :

```
{
    "ImageId": "ami-dfc39aef",
    "KeyName": "mykey",
    "SecurityGroups": [
        "my-sg"
    ],
    "InstanceType": "t2.micro",
    "Monitoring": {
        "Enabled": true
    }
}
```

YAML

Contenu du fichier YAML :

```
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
- 'my-sg'
InstanceType: 't2.micro'
Monitoring:
  Enabled: true
```

L'exemple suivant utilise le fichier d'entrée en combinaison avec le `--dry-run` paramètre pour exécuter à sec la commande afin de confirmer si vous disposez des autorisations requises et si vous avez rempli le fichier avec des valeurs valides.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --dry-run
```

```
A client error (DryRunOperation) occurred when calling the RunInstances operation:
 Request would have succeeded, but DryRun flag is set.
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --dry-run
```

```
A client error (DryRunOperation) occurred when calling the RunInstances operation:
 Request would have succeeded, but DryRun flag is set.
```

L'exemple suivant utilise ensuite le même fichier d'entrée, mais avec le `--no-dry-run` paramètre permettant d'exécuter la commande dans son intégralité.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --no-dry-run --
output json
```

```
{
    "OwnerId": "123456789012",
    "ReservationId": "r-d94a2b1",
    "Groups": [],
    "Instances": [
...
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --no-dry-run --
output yaml
```

```
OwnerId: '123456789012'
ReservationId: 'r-d94a2b1',
Groups":
- ''
Instances:
...
```

En utilisant la syntaxe abrégée dans le AWS CLI

Le AWS Command Line Interface (AWS CLI) peut accepter plusieurs de ses paramètres d'option au format JSON. Toutefois, il peut être fastidieux de saisir de longues listes ou structures JSON sur la ligne de commande. Pour vous faciliter la tâche, il prend AWS CLI également en charge une syntaxe abrégée qui permet une représentation plus simple des paramètres de vos options qu'en utilisant le format JSON complet.

Rubriques

- Paramètres de structure avec paires clé-valeur
- Chargement d'un fichier en tant que valeur de syntaxe abrégée
- En utilisant une syntaxe abrégée avec AWS CLI

Paramètres de structure avec paires clé-valeur

La syntaxe abrégée utilisée dans le AWS CLI permet aux utilisateurs de saisir plus facilement des paramètres plats (structures non imbriquées). Le format est une liste séparée par des virgules des paires clé-valeur. Veillez à utiliser les règles de citation et d'échappement adaptées à votre terminal, car les syntaxes abrégées sont des chaînes.

Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

Est équivalent à l'exemple suivant, formaté en JSON.

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

Windows

```
--option "key1=value1,key2=value2,key3=value3"
```

Est équivalent à l'exemple suivant, formaté en JSON.

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

Il ne doit y avoir aucun espace entre chaque paire clé-valeur séparée par des virgules. Voici un exemple de commande Amazon `update-table` DynamoDB avec l'option spécifiée en `--provisioned-throughput` abrégé.

```
$ aws dynamodb update-table \
    --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \
    --table-name MyDDBTable
```

Cela équivaut à l'exemple suivant au format JSON.

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \
    --table-name MyDDBTable
```

Chargement d'un fichier en tant que valeur de syntaxe abrégée

Lorsqu'une valeur est grande ou complexe, il est souvent plus facile de la charger en tant que valeur. Pour charger un fichier en tant que valeur de syntaxe abrégée, le formatage changera légèrement. Au lieu de cela, `key=value` vous utiliserez l'@=opérateur au lieu de l'=opérateur. Cela @= signifie AWS CLI que la valeur doit être lue comme un chemin de fichier et non comme une chaîne. Lorsque vous chargez des fichiers en syntaxe abrégée, les règles habituelles de formatage des AWS CLI fichiers s'appliquent. L'exemple suivant montre une paire clé-valeur chargeant un fichier pour sa valeur.

Linux or macOS

```
--option key@=file://template.txt
```

Windows

```
--option "key1@=file://template.txt"
```

L'exemple suivant montre comment charger un fichier de certificat pour la `aws rolesanywhere create-trust-anchor` commande.

```
$ aws rolesanywhere create-trust-anchor --name TrustAnchor \
    --source sourceData={x509CertificateData@=file://root-ca.crt},sourceType="CERTIFICATE_BUNDLE"  \
    --enabled
```

En utilisant une syntaxe abrégée avec AWS CLI

Vous pouvez spécifier des paramètres d'entrée d'une liste peuvent être spécifiés de deux manières : JSON ou syntaxe raccourcie. La syntaxe raccourcie de l' AWS CLI est conçue pour faciliter la transmission de listes comportant des nombres, des chaînes ou des structures non imbriquées.

Le format de base est illustré ici, avec les valeurs de la liste séparées par un espace simple.

```
--option value1 value2 value3
```

Cela équivaut à l'exemple suivant au format JSON.

```
--option '[value1,value2,value3]'
```

Comme indiqué plus haut, vous pouvez utiliser une syntaxe raccourcie pour indiquer une liste de nombres, une liste de chaînes ou une liste de structures non imbriquées. Voici un exemple de `stop-instances` commande pour Amazon Elastic Compute Cloud (Amazon EC2), où le paramètre d'entrée (liste de chaînes) de l'`--instance-ids`option est spécifié sous forme abrégée.

```
$ aws ec2 stop-instances \
    --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

Cela équivaut à l'exemple suivant au format JSON.

```
$ aws ec2 stop-instances \
    --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

L'exemple suivant montre la EC2 `create-tags` commande Amazon, qui prend une liste de structures non imbriquées pour l'`--tags`option. L'option `--resources` spécifie l'ID de l'instance à baliser.

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2
 Key=My3rdTag,Value=Value3
```

Cela équivaut à l'exemple suivant au format JSON. Le paramètre JSON est écrit sur plusieurs lignes pour une meilleure lisibilité.

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags '[
        {"Key": "My1stTag", "Value": "Value1"},
        {"Key": "My2ndTag", "Value": "Value2"},
        {"Key": "My3rdTag", "Value": "Value3"}
    ]'
```

Activation et utilisation des invites de commande dans le AWS CLI

La AWS CLI version 2 peut vous demander des commandes, des paramètres et des ressources lorsque vous exécutez une `aws` commande.

Rubriques

- Comment ça marche
- Fonctionnalités d'invite automatique
- Modes d'invite automatique
- Configurer l'invite automatique

Comment ça marche

Si cette option est activée, l'invite automatique vous permet d'utiliser la touche ENTER pour terminer une commande partiellement saisie. Après avoir appuyé sur la touche ENTER, des commandes, des paramètres et des ressources sont suggérés en fonction de ce que vous continuez à taper. Les suggestions indiquent le nom de la commande, du paramètre ou de la ressource sur la gauche et leur

description sur la droite. Pour sélectionner et utiliser une suggestion, utilisez les flèches pour surligner une ligne, puis appuyez sur la touche ESPACE. Lorsque vous avez fini de saisir votre commande, appuyez sur ENTER pour l'utiliser. L'exemple suivant montre à quoi ressemble une liste suggérée à partir d'une invite automatique.

```
$ aws
> aws a
       accessanalyzer               Access Analyzer
       acm                          AWS Certificate Manager
       acm-pca                      AWS Certificate Manager Private Certificate
 Authority
       alexaforbusiness             Alexa For Business
       amplify                      AWS Amplify
```

Fonctionnalités d'invite automatique

L'invite automatique contient les fonctionnalités utiles suivantes :

Panneau de documentation

Fournit la documentation d'aide pour la commande en cours. Pour ouvrir la documentation, appuyez sur la touche F3.

Saisie automatique de commandes

`aws`Suggère des commandes à utiliser. Pour afficher une liste, entrez partiellement la commande. L'exemple suivant recherche un service commençant par la lettre`a`.

```
$ aws
> aws a
       accessanalyzer               Access Analyzer
       acm                          AWS Certificate Manager
       acm-pca                      AWS Certificate Manager Private Certificate
 Authority
       alexaforbusiness             Alexa For Business
       amplify                      AWS Amplify
```

Complétion des paramètres

Une fois qu'une commande est saisie, l'invite automatique démarre pour suggérer des paramètres. Les descriptions des paramètres incluent le type de valeur et une description de ce qu'est le paramètre. Les paramètres requis sont répertoriés en premier et sont étiquetés comme

requis. L'exemple suivant montre la liste automatique des paramètres pour`aws dynamodb describe-table`.

```
$ aws dynamodb describe-table
> aws dynamodb describe-table
                              --table-name (required)  [string] The name of the
 table to describe.
                               --cli-input-json         [string] Reads arguments
 from the JSON string provided. The JSON string follows the format provide...
                               --cli-input-yaml         [string] Reads arguments
 from the YAML string provided. The YAML string follows the format provide...
                               --generate-cli-skeleton  [string] Prints a JSON
 skeleton to standard output without sending an API request. If provided wit...
```

Achèvement des ressources

L'invite automatique effectue des appels d'API AWS en utilisant les propriétés des ressources AWS disponibles pour suggérer des valeurs de ressources. Cela permet à l'invite automatique de suggérer d'éventuelles ressources que vous possédez lors de la saisie des paramètres. Dans l'exemple suivant, l'invite automatique répertorie les noms de vos tables lorsque vous saisissez le `--table-name` paramètre de la `aws dynamodb describe-table` commande.

```
$ aws dynamodb describe-table
> aws dynamodb describe-table --table-name
                                            Table1
                                            Table2
                                            Table3
```

Achèvement de la sténographie

Pour les paramètres utilisant une syntaxe abrégée, l'invite automatique suggère les valeurs à utiliser. Dans l'exemple suivant, l'invite automatique répertorie les valeurs de syntaxe abrégée pour le `--placement` paramètre de la commande. `aws ec2 run-instances`

```
$ aws ec2 run-instances
> aws ec2 run-instances --placement
 AvailabilityZone=      [string] The Availability Zone of the instance. If not
 specified, an Availability Zone wil...
 Affinity=              [string] The affinity setting for the instance on the
 Dedicated Host. This parameter is no...
 GroupName=             [string] The name of the placement group the instance is in.
```

```
 PartitionNumber=      [integer] The number of the partition the instance is in.
 Valid only if the placement grou...
```

Achèvement du fichier

Lorsque vous renseignez des paramètres dans `aws` des commandes, la saisie automatique suggère des noms de fichiers locaux après avoir utilisé le préfixe ou. `file://` `fileb://` Dans l'exemple suivant, l'invite automatique suggère des fichiers locaux après avoir saisi `--item file://` la `aws ec2 run-instances` commande.

```
$ aws ec2 run-instances
> aws ec2 run-instances --item file://
                         item1.txt
                         file1.json
                         file2.json
```

Achèvement de la région

Lorsque vous utilisez le paramètre global`--region`, l'invite automatique répertorie les régions parmi lesquelles sélectionner. Dans l'exemple suivant, l'invite automatique suggère les régions par ordre alphabétique après avoir saisi `--region` la `aws dynamodb list-tables` commande.

```
$ aws dynamodb list-tables
> aws dynamodb list-tables --region
                                    af-south-1
                                    ap-east-1
                                    ap-northeast-1
                                    ap-northeast-2
```

Complétion du profil

Lorsque vous utilisez le paramètre global`--profile`, l'invite automatique répertorie vos profils. Dans l'exemple suivant, l'invite automatique suggère vos profils après avoir saisi `--profile` la `aws dynamodb list-tables` commande.

```
$ aws dynamodb list-tables
> aws dynamodb list-tables --profile
                                    profile1
                                    profile2
                                    profile3
```

Recherche floue

Commandes et valeurs complètes contenant un jeu de caractères spécifique. Dans l'exemple suivant, l'invite automatique suggère les régions qui contiennent une `eu` fois `--region eu` la `aws dynamodb list-tables` commande saisie.

```
$ aws dynamodb list-tables
> aws dynamodb list-tables --region west
                                        eu-west-1
                                        eu-west-2
                                        eu-west-3
                                        us-west-1
```

Historique

Pour afficher et exécuter les commandes précédemment utilisées en mode invite automatique, appuyez sur CTRL+R. L'historique répertorie les commandes précédentes que vous pouvez sélectionner à l'aide des touches fléchées. Dans l'exemple suivant, l'historique du mode d'invite automatique est affiché.

```
$ aws
> aws
        dynamodb list-tables
        s3 ls
```

Modes d'invite automatique

L'invite automatique pour la AWS CLI version 2 dispose de 2 modes qui peuvent être configurés :

- Mode complet : utilise l'invite automatique chaque fois que vous essayez d'exécuter une `aws` commande, que vous l'appeliez manuellement à l'aide du `--cli-auto-prompt` paramètre ou que vous l'activiez définitivement. Cela inclut d'appuyer sur ENTER après une commande complète ou une commande incomplète.
- Mode partiel : utilise l'invite automatique si une commande est incomplète ou ne peut pas être exécutée en raison d'erreurs de validation côté client. Ce mode est particulièrement utile si vous avez des scripts ou des runbooks préexistants, ou si vous souhaitez uniquement être automatiquement invité à saisir des commandes que vous ne connaissez pas au lieu d'être invité à chaque commande.

Configurer l'invite automatique

Pour configurer l'invite automatique, vous pouvez utiliser les méthodes suivantes par ordre de priorité :

- Les options de ligne de commande activent ou désactivent l'invite automatique pour une seule commande. `--cli-auto-prompt`À utiliser pour appeler l'invite automatique et `--no-cli-auto-prompt` pour désactiver l'invite automatique.
- Les variables d'environnement utilisent la `aws_cli_auto_prompt` variable.
- Les fichiers de configuration partagés utilisent ce `cli_auto_prompt` paramètre.

Contrôle de la sortie des commandes dans le AWS CLI

Cette section décrit les différentes manières de contrôler la sortie de AWS Command Line Interface (AWS CLI). La personnalisation de la AWS CLI sortie dans votre terminal peut améliorer la lisibilité, rationaliser l'automatisation des scripts et faciliter la navigation dans des ensembles de données plus volumineux.

Il AWS CLI prend en charge plusieurs formats de sortie `json`, notamment `text`, `yaml`, et `table`. Certains services proposent une pagination côté serveur pour leurs données et AWS CLI fournissent leurs propres fonctionnalités côté client pour des options de pagination supplémentaires.

Enfin, il AWS CLI dispose d'un filtrage côté serveur et côté client que vous pouvez utiliser individuellement ou ensemble pour filtrer votre sortie. AWS CLI

Rubriques

- Sortie sensible
- Options de sortie côté serveur ou côté client
- Réglage du format de sortie dans AWS CLI
- À l'aide des options de pagination du AWS CLI
- Filtrage de la sortie dans le AWS CLI

Sortie sensible

Certaines opérations AWS CLI peuvent renvoyer des informations pouvant être considérées comme sensibles, notamment des informations provenant de variables d'environnement. L'exposition de

ces informations peut représenter un risque de sécurité dans certains scénarios ; par exemple, les informations peuvent être incluses dans les journaux d'intégration continue et de déploiement continu (CI/CD). Il est donc important de vérifier à quel moment vous incluez une telle sortie dans vos journaux et de supprimer la sortie lorsqu'elle n'est pas nécessaire.

Pour plus d'informations sur la protection des données sensibles, consultezthe section called “Protection des données”.

Tenez compte des meilleures pratiques suivantes :

- Envisagez de récupérer vos secrets par programmation dans un magasin de secrets, tel que. AWS Secrets Manager
- Vérifiez le contenu de vos journaux de compilation pour vous assurer qu'ils ne contiennent pas d'informations sensibles. Envisagez des approches telles que le transfert `/dev/null` ou la capture de la sortie sous forme de bash ou de PowerShell variable pour supprimer les sorties de commande.

  Voici un exemple de bash pour rediriger la sortie, mais pas les erreurs, vers : `/dev/null`

  ```
  $ aws s3 ls > /dev/null
  ```

  Pour plus d'informations sur la suppression de la sortie de votre terminal, consultez la documentation utilisateur du terminal que vous utilisez.
- Tenez compte de l'accès à vos journaux et délimitez cet accès en fonction de votre cas d'utilisation.

Options de sortie côté serveur ou côté client

AWS CLI Il dispose d'un filtrage côté serveur et côté client que vous pouvez utiliser individuellement ou ensemble pour filtrer votre sortie. AWS CLI Le filtrage côté serveur est traité en premier et renvoie votre sortie pour le filtrage côté client. Le filtrage côté serveur est pris en charge par l'API du service. Le filtrage côté client est pris en charge par le AWS CLI client à l'aide du `--query` paramètre.

Les options de sortie côté serveur sont des fonctionnalités directement prises en charge par l' Service AWS API. Les données filtrées ou renvoyées ne sont pas envoyées au client, ce qui peut accélérer les temps de réponse HTTP et améliorer la bande passante pour les ensembles de données plus volumineux.

Les options de sortie côté client sont des fonctionnalités créées par le. AWS CLI Toutes les données sont envoyées au client, puis les AWS CLI filtres ou les pages affichent le contenu. Les opérations côté client ne permettent pas d'économiser de la vitesse ou de la bande passante pour les ensembles de données plus volumineux.

Lorsque les options côté serveur et côté client sont utilisées conjointement, les opérations côté serveur sont d'abord effectuées, puis envoyées au client pour les opérations côté client. Cela utilise les économies de vitesse et de bande passante potentielles des options côté serveur, tout en utilisant des AWS CLI fonctionnalités supplémentaires pour obtenir le résultat souhaité.

Réglage du format de sortie dans AWS CLI

Cette rubrique décrit les différents formats de sortie pour le AWS Command Line Interface (AWS CLI). AWS CLI Supporte les formats de sortie suivants :

- **`json`**— La sortie est formatée sous forme de chaîne JSON.
- **`yaml`**— La sortie est formatée sous forme de chaîne YAML.
- **`yaml-stream`**— La sortie est diffusée et formatée sous forme de chaîne YAML. Le streaming permet de traiter plus rapidement de gros types de données.
- **`text`**— La sortie est formatée sous la forme de plusieurs lignes de valeurs de chaîne séparées par des tabulations. Cela peut être utile pour transmettre le résultat à un processeur de texte, comme `grep`ou `sed`, ou `awk`.
- **`table`**— La sortie est formatée sous forme de tableau en utilisant les caractères +|- pour former les bordures des cellules. La présentation des informations est dans un format beaucoup plus lisible par l'utilisateur que les autres, mais peu pratique du point de vue programmation.

Comment sélectionner le format de sortie

Comme expliqué dans la rubrique de configuration, le format de sortie peut être spécifié de trois façons :

- Utilisation de l'**`output`**option dans un profil nommé dans le **`config`** fichier — L'exemple suivant définit le format de sortie par défaut sur`text`.

```
[default]
output=text
```

- Utilisation de la variable d'**`AWS_DEFAULT_OUTPUT`**environnement — La sortie suivante définit le format des commandes de `table` cette session de ligne de commande jusqu'à ce que la variable soit modifiée ou que la session se termine. L'utilisation de cette variable d'environnement remplace toute valeur définie dans le fichier `config`.

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- Utilisation de l'**`--output`**option sur la ligne de commande — L'exemple suivant définit la sortie de cette seule commande sur`json`. Utiliser cette option sur la commande remplace n'importe quelle variable d'environnement actuellement définie ou la valeur dans le fichier `config`.

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

> **Important**
>
> Le type de sortie que vous spécifiez modifie le mode de fonctionnement de l'`--query`option :
>
> - Si vous le spécifiez`--output text`, la sortie est paginée avant que le `--query` filtre ne soit appliqué, et la AWS CLI requête est exécutée une fois sur chaque page de la sortie. De ce fait, la requête inclut le premier élément correspondant sur chaque page, ce qui peut entraîner une sortie supplémentaire inattendue. Pour filtrer également la sortie, vous pouvez utiliser d'autres outils de ligne de commande tels que `head` `outail`.
> - Si vous spécifiez `--output json` `--output yaml`, ou si `--output yaml-stream` la sortie est entièrement traitée comme une structure native unique avant que le `--query` filtre ne soit appliqué. La requête n'est AWS CLI exécutée qu'une seule fois sur l'ensemble de la structure, produisant un résultat filtré qui est ensuite affiché.

Format de sortie JSON

JSON est le format de sortie par défaut de l' AWS CLI. La plupart des langages de programmation peuvent facilement décoder les chaînes JSON à l'aide de fonctions intégrées ou de bibliothèques disponibles au public. Vous pouvez combiner la sortie JSON avec l'option --query de manière puissante pour filtrer et formater la sortie au format AWS CLI JSON.

Pour un filtrage plus avancé que vous ne pourriez peut-être pas faire avec `--query`, vous pouvez envisager `jq`, processeur JSON de ligne de commande. Vous pouvez le télécharger et trouver le didacticiel officiel à l'adresse http://stedolan.github.io/jq/.

Voici un exemple de données JSON :

```
$ aws iam list-users --output json
```

```
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Admin",
            "UserId": "AIDA1111111111EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Admin",
            "CreateDate": "2014-10-16T16:03:09+00:00",
            "PasswordLastUsed": "2016-06-03T18:37:29+00:00"
        },
        {
            "Path": "/backup/",
            "UserName": "backup-user",
            "UserId": "AIDA2222222222EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/backup/backup-user",
            "CreateDate": "2019-09-17T19:30:40+00:00"
        },
        {
            "Path": "/",
            "UserName": "cli-user",
            "UserId": "AIDA3333333333EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/cli-user",
            "CreateDate": "2019-09-17T19:11:39+00:00"
        }
    ]
}
```

Format de sortie YAML

YAML est un bon choix pour gérer la sortie par programmation avec des services et des outils qui émettent ou consomment des chaînes au format YAML, comme AWS CloudFormation sa prise en charge des modèles au format YAML.

Pour un filtrage plus avancé que vous ne pourriez peut-être pas faire avec `--query`, vous pouvez envisager `yq`, processeur YAML de ligne de commande. Vous pouvez le télécharger `yq` dans le référentiel yq sur GitHub.

Voici un exemple de résultat YAML :.

```
$ aws iam list-users --output yaml
```

```
Users:
- Arn: arn:aws:iam::123456789012:user/Admin
  CreateDate: '2014-10-16T16:03:09+00:00'
  PasswordLastUsed: '2016-06-03T18:37:29+00:00'
  Path: /
  UserId: AIDA1111111111EXAMPLE
  UserName: Admin
- Arn: arn:aws:iam::123456789012:user/backup/backup-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /backup/
  UserId: AIDA2222222222EXAMPLE
  UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
- Arn: arn:aws:iam::123456789012:user/cli-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /
  UserId: AIDA3333333333EXAMPLE
  UserName: cli-user
```

Format de sortie de flux YAML

Le `yaml-stream` format tire parti du format YAML tout en offrant une visualisation plus réactive/plus rapide de grands ensembles de données en vous transmettant les données en streaming. Vous pouvez commencer à visualiser et à utiliser les données YAML avant le téléchargement complet de la requête.

Pour un filtrage plus avancé que vous ne pourriez peut-être pas faire avec `--query`, vous pouvez envisager `yq`, processeur YAML de ligne de commande. Vous pouvez le télécharger `yq` dans le référentiel yq sur GitHub.

Voici un exemple de résultat `yaml-stream`.

```
$ aws iam list-users --output yaml-stream
```

```
- IsTruncated: false
  Users:
  - Arn: arn:aws:iam::123456789012:user/Admin
    CreateDate: '2014-10-16T16:03:09+00:00'
    PasswordLastUsed: '2016-06-03T18:37:29+00:00'
    Path: /
    UserId: AIDA1111111111EXAMPLE
    UserName: Admin
  - Arn: arn:aws:iam::123456789012:user/backup/backup-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /backup/
    UserId: AIDA2222222222EXAMPLE
    UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
  - Arn: arn:aws:iam::123456789012:user/cli-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /
    UserId: AIDA3333333333EXAMPLE
    UserName: cli-user
```

Voici un exemple de `yaml-stream` sortie associé à l'utilisation du `--page-size` paramètre pour paginer le contenu YAML diffusé en continu.

```
$ aws iam list-users --output yaml-stream --page-size 2
```

```
- IsTruncated: true
  Marker: ab1234cdef5ghi67jk8lmo9p/
q012rs3t445uv6789w0x1y2z/345a6b78c9d00/1efgh234ij56klmno78pqrstu90vwxyx
  Users:
  - Arn: arn:aws:iam::123456789012:user/Admin
    CreateDate: '2014-10-16T16:03:09+00:00'
    PasswordLastUsed: '2016-06-03T18:37:29+00:00'
    Path: /
    UserId: AIDA1111111111EXAMPLE
    UserName: Admin
  - Arn: arn:aws:iam::123456789012:user/backup/backup-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /backup/
    UserId: AIDA2222222222EXAMPLE
    UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
- IsTruncated: false
  Users:
  - Arn: arn:aws:iam::123456789012:user/cli-user
```

```
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /
    UserId: AIDA3333333333EXAMPLE
    UserName: cli-user
```

Format de sortie texte

Le `text` format organise la AWS CLI sortie en lignes délimitées par des tabulations. Il fonctionne bien avec les outils de texte Unix traditionnels tels que `grep`,`sed`,`awk`, et le traitement de texte effectué par PowerShell.

Le format de sortie `text` suit la structure de base ci-dessous. Les colonnes sont triées dans l'ordre alphabétique sur les noms de clés correspondants de l'objet JSON sous-jacent.

```
IDENTIFIER  sorted-column1 sorted-column2
IDENTIFIER2 sorted-column1 sorted-column2
```

Voici un exemple de résultat `text`. Chaque champ est un onglet séparé des autres, avec un onglet supplémentaire où il y a un champ vide.

```
$ aws iam list-users --output text
```

```
USERS   arn:aws:iam::123456789012:user/Admin                2014-10-16T16:03:09+00:00
 2016-06-03T18:37:29+00:00   /          AIDA1111111111EXAMPLE   Admin
USERS   arn:aws:iam::123456789012:user/backup/backup-user   2019-09-17T19:30:40+00:00
                            /backup/   AIDA2222222222EXAMPLE   backup-user
USERS   arn:aws:iam::123456789012:user/cli-user             2019-09-17T19:11:39+00:00
                            /          AIDA3333333333EXAMPLE   cli-user
```

La quatrième colonne est le champ `PasswordLastUsed`. Elle est vide pour les deux dernières entrées, car ces utilisateurs ne se connectent jamais à la console AWS Management Console.

> **Important**
>
> Si vous spécifiez la sortie *`text`*, il est également fortement recommandé de toujours utiliser l'option *`--query`* pour assurer un comportement cohérent.
> Cela est dû au fait que le format de texte classe les colonnes de sortie par ordre alphabétique selon le nom de clé de l'objet JSON sous-jacent renvoyé par le AWS service, et des ressources similaires peuvent ne pas porter les mêmes noms de clé. Par exemple, la

représentation JSON d'une instance EC2 Amazon basée sur Linux peut comporter des éléments qui ne sont pas présents dans la représentation JSON d'une instance Windows, ou vice versa. En outre, des ressources peuvent comporter des éléments clé-valeur ajoutés ou supprimés lors de futures mises à jour, ce qui modifie l'ordre des colonnes. C'est à ce stade que le code `--query` augmente la fonctionnalité de la sortie `text` afin de vous offrir un contrôle total sur le format de sortie.
Dans l'exemple suivant, la commande spécifie les éléments à afficher et définit l'ordre des colonnes avec la notation de liste `[key1, key2, ...]`. Cela vous permet d'être absolument sûr que les valeurs de clés correctes sont toujours affichées dans la colonne prévue. Enfin, remarquez comment les AWS CLI sorties sont `None` des valeurs pour des clés qui n'existent pas.

```
$ aws iam list-users --output text --query 'Users[*].
[UserName,Arn,CreateDate,PasswordLastUsed,UserId]'
```

```
Admin         arn:aws:iam::123456789012:user/Admin
 2014-10-16T16:03:09+00:00   2016-06-03T18:37:29+00:00   AIDA1111111111EXAMPLE
backup-user   arn:aws:iam::123456789012:user/backup-user
 2019-09-17T19:30:40+00:00   None                        AIDA2222222222EXAMPLE
cli-user      arn:aws:iam::123456789012:user/cli-backup
 2019-09-17T19:11:39+00:00   None                        AIDA3333333333EXAMPLE
```

L'exemple suivant montre comment vous pouvez utiliser `grep` et `awk` avec la sortie `text` à partir de la commande `aws ec2 describe-instances`. La première commande affiche la zone de disponibilité, l'état actuel et l'ID d'instance de chaque instance dans la sortie `text`. La deuxième commande traite cette sortie pour afficher uniquement l'instance IDs de toutes les instances en cours d'exécution dans la zone de `us-west-2a` disponibilité.

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text
```

```
us-west-2a      running i-4b41a37c
us-west-2a      stopped i-a071c394
us-west-2b      stopped i-97a217a0
us-west-2a      running i-3045b007
us-west-2a      running i-6fc67758
```

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text | grep us-west-2a |
 grep running | awk '{print $3}'
```

```
i-4b41a37c
i-3045b007
i-6fc67758
```

L'exemple suivant va encore plus loin et indique comment filtrer la sortie, mais également comment utiliser cette sortie pour automatiser la modification des types d'instance pour chaque instance arrêtée.

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].[State.Name,
 InstanceId]' --output text |
> grep stopped |
> awk '{print $2}' |
> while read line;
> do aws ec2 modify-instance-attribute --instance-id $line --instance-type '{"Value":
 "m1.medium"}';
> done
```

La `text` sortie peut également être utile dans PowerShell. Comme les colonnes de la `text` sortie sont séparées par des tabulations, vous pouvez facilement diviser la sortie en tableau en utilisant le PowerShell `` `t `` délimiteur. La commande suivante affiche la valeur de la troisième colonne (`InstanceId`) si la première colonne (`AvailabilityZone`) correspond à la chaîne `us-west-2a`.

```
PS C:\>aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text |
%{if ($_.split("`t")[0] -match "us-west-2a") { $_.split("`t")[2]; } }
```

```
-4b41a37c
i-a071c394
i-3045b007
i-6fc67758
```

Notez que même si l'exemple précédent montre comment utiliser le `--query` paramètre pour analyser les objets JSON sous-jacents et extraire la colonne souhaitée, il PowerShell possède sa propre capacité à gérer le JSON, si la compatibilité entre plateformes n'est pas un problème. Au lieu de gérer la sortie sous forme de texte, comme l'exigent la plupart des interfaces de commande,

PowerShell vous pouvez utiliser l'`ConvertFrom-JSO`Napplet de commande pour produire un objet structuré hiérarchiquement. Vous pouvez ensuite accéder directement au membre souhaité à partir de cet objet.

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

> **Tip**
>
> Si vous avez une sortie de texte et que vous filtrez la sortie vers un champ unique à l'aide du paramètre `--query`, la sortie consiste en une seule ligne de valeurs séparées par une tabulation. Pour voir chaque valeur sur une ligne distincte, vous pouvez placer le champ de sortie entre crochets comme illustré dans les exemples suivants :
> Séparées par tabulation, sortie à ligne unique :
>
> ```
> $ aws iam list-groups-for-user --user-name susan  --output text --query
>  "Groups[].GroupName"
> ```
>
> ```
> HRDepartment    Developers      SpreadsheetUsers  LocalAdmins
> ```
>
> Chaque valeur sur sa propre ligne en plaçant `[GroupName]` entre crochets :
>
> ```
> $ aws iam list-groups-for-user --user-name susan  --output text --query
>  "Groups[].[GroupName]"
> ```
>
> ```
> HRDepartment
> Developers
> SpreadsheetUsers
> LocalAdmins
> ```

Format de sortie de tableau

Le format `table` produit des représentations explicites de la sortie complexe de l' AWS CLI sous la forme d'un tableau.

```
$ aws iam list-users --output table
```

```
-----------------------------------------------------------------------------------------------
|
 ListUsers                                                                    |
+-----------------------------------------------------------------------------------------------
+
||
 Users                                                                    ||
|+-----------------------------------------------------+---------------------------
+---------------------------+----------+----------------------+-------------+|
||                        Arn                        |       CreateDate          |
 PasswordLastUsed       |   Path   |        UserId         |   UserName  ||
|+-----------------------------------------------------+---------------------------
+---------------------------+----------+----------------------+-------------+|
||  arn:aws:iam::123456789012:user/Admin              | 2014-10-16T16:03:09+00:00 |
 2016-06-03T18:37:29+00:00 | /        | AIDA1111111111EXAMPLE | Admin       ||
||  arn:aws:iam::123456789012:user/backup/backup-user | 2019-09-17T19:30:40+00:00 |
                       | /backup/ | AIDA2222222222EXAMPLE | backup-user ||
||  arn:aws:iam::123456789012:user/cli-user           | 2019-09-17T19:11:39+00:00 |
                       | /        | AIDA3333333333EXAMPLE | cli-user    ||
+-----------------------------------------------------------------------------------------------
+
```

Vous pouvez combiner l'option `--query` avec le format `table` pour afficher un ensemble d'éléments pré-sélectionnés dans la sortie brute. Notez les différences de sortie entre le dictionnaire et les notations de liste : les noms de colonnes sont triés par ordre alphabétique dans le premier exemple et les colonnes sans nom sont triées en fonction du choix de l'utilisateur dans le second exemple. Pour plus d'informations sur l'option `--query`, consultez Filtrage de la sortie dans le AWS CLI.

```
$ aws ec2 describe-volumes --query 'Volumes[*].
{ID:VolumeId,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}' --
output table
```

```
-----------------------------------------------------
|                  DescribeVolumes                  |
+------------+----------------+--------------+-------+
|     AZ     |      ID        | InstanceId   | Size  |
+------------+----------------+--------------+-------+
|  us-west-2a|  vol-e11a5288  |  i-a071c394  |  30   |
|  us-west-2a|  vol-2e410a47  |  i-4b41a37c  |  8    |
+------------+----------------+--------------+-------+
```

```
$ aws ec2 describe-volumes --query 'Volumes[*].
[VolumeId,Attachments[0].InstanceId,AvailabilityZone,Size]' --output table
```

```
-------------------------------------------------
|                 DescribeVolumes                |
+--------------+-------------+--------------+-----+
|  vol-e11a5288|  i-a071c394  |  us-west-2a  |  30 |
|  vol-2e410a47|  i-4b41a37c  |  us-west-2a  |  8  |
+--------------+-------------+--------------+-----+
```

À l'aide des options de pagination du AWS CLI

Cette rubrique décrit les différentes méthodes de pagination de la sortie à partir de AWS Command Line Interface (AWS CLI).

Il existe principalement deux manières de contrôler la pagination à partir du AWS CLI.

- Utilisation des paramètres de pagination côté serveur.
- Utilisation de votre programme de pagination côté client en sortie par défaut.

Les paramètres de pagination côté serveur traitent en premier et toute sortie est envoyée à la pagination côté client.

Pagination côté serveur

Pour la plupart des commandes qui renvoient une longue liste d'éléments, plusieurs options côté serveur permettent de contrôler le nombre d'éléments inclus dans la sortie lorsque l'API d' AWS CLI un service est appelée pour remplir la liste. AWS CLI La pagination côté serveur dans le AWS CLI est activée par l'API du AWS service. Par conséquent, ces options ne fonctionnent que si l'API du service les active.

Les options de la plupart des AWS CLI commandes sont les suivantes :

- Comment utiliser le paramètre --no-paginate
- Comment utiliser le paramètre --page-size
- Comment utiliser le paramètre --max-items
- Comment utiliser le paramètre --starting-token

Par défaut, il AWS CLI utilise un format de page déterminé par le service individuel et récupère tous les éléments disponibles. Par exemple, Amazon S3 a une taille de page par défaut de 1 000. Si vous exécutez `aws s3api list-objects` un paquet Amazon S3 contenant 3 500 objets, il passe AWS CLI automatiquement quatre appels à Amazon S3, gérant la logique de pagination spécifique au service pour vous en arrière-plan et renvoyant les 3 500 objets dans le résultat final.

Pour savoir si une commande spécifique possède une pagination côté serveur, consultez le guide de référence AWS CLI version 2 du .

Comment utiliser le paramètre --no-paginate

L'`--no-paginate`option désactive les jetons de pagination suivants côté client. Lorsque vous utilisez une commande, par défaut, elle effectue AWS CLI automatiquement plusieurs appels pour renvoyer tous les résultats possibles afin de créer une pagination. Un appel par page. La désactivation de la pagination AWS CLI ne nécessite qu'une seule fois la première page des résultats de commande.

Par exemple, si vous exécutez `aws s3api list-objects` un bucket Amazon S3 contenant 3 500 objets, AWS CLI seul le premier appel est envoyé à Amazon S3, renvoyant uniquement les 1 000 premiers objets dans le résultat final.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --no-paginate
{
    "Contents": [
...
```

Comment utiliser le paramètre --page-size

Si vous rencontrez des problèmes lors de l'exécution de commandes de liste sur un grand nombre de ressources, la taille de page par défaut est peut-être trop élevée. Cela peut entraîner un dépassement de la durée maximale autorisée pour les appels aux AWS services et générer une erreur de « expiration du délai imparti ». Vous pouvez utiliser cette `--page-size` option pour spécifier que le AWS CLI nombre d'éléments demandés à chaque appel au AWS service est réduit. Il récupère AWS CLI toujours la liste complète, mais exécute un plus grand nombre d'appels d'API de service en arrière-plan et récupère un plus petit nombre d'éléments à chaque appel. Cela donne aux appels individuels une plus grande chance de réussite sans délai d'expiration. Modifier la taille de la page n'affecte pas la sortie ; cela affecte uniquement le nombre d'appels d'API qui doivent être effectués pour générer la sortie.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --page-size 100
{
    "Contents": [
...
```

Comment utiliser le paramètre --max-items

Pour inclure moins d'éléments à la fois dans la AWS CLI sortie, utilisez l'`--max-items`option. Le gère AWS CLI toujours la pagination avec le service comme décrit précédemment, mais n'imprime que le nombre d'éléments que vous spécifiez à la fois.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --max-items 100
{
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==",
    "Contents": [
...
```

Comment utiliser le paramètre --starting-token

Si le nombre d'éléments en sortie (`--max-items`) est inférieur au nombre total d'éléments renvoyés par les appels d'API sous-jacents, la sortie inclut un code `NextToken` que vous pouvez transmettre dans une commande suivante pour extraire le prochain ensemble d'éléments. L'exemple suivant montre comment utiliser la valeur `NextToken` renvoyée par l'exemple précédent et vous permet de récupérer la deuxième centaine d'éléments.

> **Note**
>
> Le paramètre `--starting-token` ne peut pas être nul ou vide. Si la commande précédente ne renvoie pas de valeur `NextToken`, il n'y a plus aucun élément à renvoyer et vous ne devez plus rappeler la commande.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --max-items 100 \
```

```
    --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==
{
    "Contents": [
...
```

Le AWS service spécifié peut ne pas retourner les articles de la même commande à chaque fois que vous appelez. Si vous spécifiez des valeurs différentes pour les codes `--page-size` et `--max-items`, vous pouvez obtenir des résultats inattendus avec des éléments manquants ou des doublons. Pour empêcher que cela ne se produise, utilisez le même nombre pour les codes `--page-size` et `--max-items`, afin de synchroniser la pagination de l' AWS CLI et celle du service sous-jacent. Vous pouvez également extraire la liste complète et exécuter les opérations de pagination nécessaires localement.

Téléavertisseur côté client

AWS CLI la version 2 fournit l'utilisation d'un programme de téléavertisseur côté client pour la sortie. Par défaut, cette fonctionnalité renvoie toutes les sorties via le programme de pager par défaut de votre système d'exploitation.

Par ordre de priorité, vous pouvez définir le pager de sortie de la manière suivante :

- En utilisant le `cli_pager` paramètre du `config` fichier dans le profil `default` ou nommé.
- À l'aide de la variable d'environnement `AWS_PAGER`.
- À l'aide de la variable d'environnement `PAGER`.

Par ordre de priorité, vous pouvez désactiver toute utilisation d'un programme de pagination externe de la manière suivante :

- Utilisez l'option de ligne de `--no-cli-pager` commande pour désactiver le pager pour une seule commande.
- Définissez le `cli_pager` paramètre ou la `AWS_PAGER` variable sur une chaîne vide.

Sujets relatifs aux pagers côté client :

- Comment utiliser le paramètre cli_pager
- Comment définir la variable d' AWS_PAGERenvironnement
- Comment utiliser l'no-cli-pageroption --
- Comment utiliser les drapeaux de téléavertisseur

Comment utiliser le paramètre cli_pager

Vous pouvez enregistrer vos paramètres de configuration utilisés fréquemment et vos informations d'identification dans les fichiers qui sont gérés par l' AWS CLI. Les paramètres d'un profil de nom ont priorité sur les paramètres du `default` profil. Pour plus d'informations sur les paramètres de configuration, consultezParamètres des fichiers de configuration et d'identification dans le AWS CLI.

L'exemple suivant définit le pager de sortie par défaut sur le `less` programme.

```
[default]
cli_pager=less
```

L'exemple suivant définit la valeur par défaut pour désactiver l'utilisation d'un pager.

```
[default]
cli_pager=
```

Comment définir la variable d' AWS_PAGERenvironnement

L'exemple suivant définit le pager de sortie par défaut sur le `less` programme. Pour plus d'informations sur les variables d'environnement, consultezConfiguration des variables d'environnement pour AWS CLI.

Linux and macOS

```
$ export AWS_PAGER="less"
```

Windows

```
C:\> setx AWS_PAGER "less"
```

Comment utiliser l'no-cli-pageroption --

Pour désactiver l'utilisation d'un téléavertisseur sur une seule commande, utilisez l'`--no-cli-pager`option. Pour plus d'informations sur les options de ligne de commande, consultezOptions de ligne de commande dans AWS CLI.

```
$ aws s3api list-objects \
```

```
    --bucket amzn-s3-demo-bucket \
    --no-cli-pager
{
    "Contents": [
...
```

Comment utiliser les drapeaux de téléavertisseur

Vous pouvez définir des drapeaux à utiliser automatiquement avec votre programme de pagination. Les drapeaux dépendent du programme de pagination que vous utilisez. Les exemples ci-dessous concernent les valeurs par défaut typiques de `less` et`more`.

Linux and macOS

Si vous ne spécifiez pas le contraire, le téléavertisseur utilisé par défaut par la AWS CLI version 2 est`less`. Si la variable d'`LESS`environnement n'est pas définie, la AWS CLI version 2 utilise les `FRX` drapeaux. Vous pouvez combiner des drapeaux en les spécifiant lors de la configuration du AWS CLI pager.

L'exemple suivant utilise le `S` drapeau. Ce drapeau se combine ensuite avec les `FRX` drapeaux par défaut pour créer un `FRXS` drapeau final.

```
$ export AWS_PAGER="less -S"
```

Si vous ne voulez aucun des `FRX` drapeaux, vous pouvez les annuler. L'exemple suivant annule le `F` drapeau pour créer un `RX` drapeau final.

```
$ export AWS_PAGER="less -+F"
```

Pour plus d'informations sur les `less` drapeaux, voir moins sur manpages.org.

Windows

Si vous ne spécifiez pas le contraire, le téléavertisseur utilisé par défaut par la AWS CLI version 2 ne `more` comporte aucun indicateur supplémentaire.

L'exemple suivant utilise le `/c` paramètre.

```
C:\> setx AWS_PAGER "more /c"
```

Pour plus d'informations sur `more` les drapeaux, consultez `more` Microsoft Docs.

Filtrage de la sortie dans le AWS CLI

Le AWS Command Line Interface (AWS CLI) dispose d'un filtrage côté serveur et côté client que vous pouvez utiliser individuellement ou ensemble pour filtrer votre sortie. AWS CLI Le filtrage côté serveur est traité en premier et renvoie votre sortie pour le filtrage côté client.

- Le filtrage côté serveur est pris en charge par l'API, et vous l'implémentez généralement avec un `--filter` paramètre. Le service renvoie uniquement des résultats correspondants, ce qui peut accélérer les temps de réponse HTTP pour les grands ensembles de données.
- Le filtrage côté client est pris en charge par le AWS CLI client à l'aide du `--query` paramètre. Ce paramètre possède des fonctionnalités que le filtrage côté serveur ne possède peut-être pas.

Rubriques

- Filtrage côté serveur
- Filtrage côté client
- Combinaison du filtrage côté serveur et côté client
- Ressources supplémentaires

Filtrage côté serveur

Le filtrage côté serveur dans le AWS CLI est fourni par l'API du AWS service. Le AWS service renvoie uniquement les enregistrements de la réponse HTTP qui correspondent à votre filtre, ce qui peut accélérer les temps de réponse HTTP pour les grands ensembles de données. Le filtrage côté serveur étant défini par l'API du service, les noms des paramètres et les fonctions varient d'un service à l'autre. Les noms de paramètres couramment utilisés pour le filtrage sont les suivants :

- `--filter`tels que mers et glace.
- `--filters`
- Noms commençant par le mot`filter`, par exemple `--filter-expression` pour la commande.

Pour savoir si une commande spécifique dispose d'un filtrage côté serveur et sur les règles de filtrage, consultez le guide de de référence.

Filtrage côté client

AWS CLI fournit des fonctionnalités intégrées de filtrage côté client basées sur JSON avec le paramètre. `--query` Le `--query` paramètre est un outil puissant que vous pouvez utiliser pour personnaliser le contenu et le style de votre sortie. Le `--query` paramètre prend la réponse HTTP qui revient du serveur et filtre les résultats avant de les afficher. Comme l'intégralité de la réponse HTTP est envoyée au client avant le filtrage, le filtrage côté client peut être plus lent que le filtrage côté serveur pour les grands ensembles de données.

Les requêtes utilisent JMESPath la syntaxe pour créer des expressions permettant de filtrer votre sortie. Pour apprendre JMESPath la syntaxe, consultez le didacticiel sur le JMESPath site Web.

> **Important**
>
> Le type de sortie que vous spécifiez modifie le mode de fonctionnement de l'`--query`option :
>
> - Si vous le spécifiez`--output text`, la sortie est paginée avant que le `--query` filtre ne soit appliqué, et la AWS CLI requête est exécutée une fois sur chaque page de la sortie. De ce fait, la requête inclut le premier élément correspondant sur chaque page, ce qui peut entraîner une sortie supplémentaire inattendue. Pour filtrer également la sortie, vous pouvez utiliser d'autres outils de ligne de commande tels que `head` `outail`.
> - Si vous spécifiez `--output json` `--output yaml`, ou si `--output yaml-stream` la sortie est entièrement traitée comme une structure native unique avant l'application du `--query` filtre. AWS CLI Exécute la requête une seule fois sur l'ensemble de la structure, produisant un résultat filtré qui est ensuite affiché.

Rubriques relatives au filtrage côté client

- Avant de commencer
- Identifiants
- Sélection dans une liste
- Filtrer les données imbriquées
- Aplatir les résultats
- Filtrage pour des valeurs spécifiques
- Expressions de tuyauterie
- Filtrage pour plusieurs valeurs d'identifiant

- Ajouter des étiquettes aux valeurs d'identification
- Fonctions
- --queryExemples avancés

Avant de commencer

> **Note**
>
> Ces exemples d'expressions de filtre sont écrits pour des shells de base de type Linux. Lorsque vous utilisez ces exemples, veillez à utiliser les bonnes règles de citation pour votre terminal shell. La façon dont votre terminal interprète vos entrées peut modifier considérablement ce qui est envoyé au. AWS CLI La façon dont votre terminal lit les guillemets simples ', les guillemets " doubles ou les backticks ` peut modifier la façon dont le contenu est lu.
> Pour de plus amples informations, veuillez consulter the section called “Citations avec chaînes”.

La sortie JSON suivante montre un exemple de ce que le `--query` paramètre peut produire. Le résultat décrit trois volumes Amazon EBS attachés à des EC2 instances Amazon distinctes.

Exemple de sortie

```
$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-17T00:55:03.000Z",
          "InstanceId": "i-a071c394",
          "VolumeId": "vol-e11a5288",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-e11a5288",
```

```
      "State": "in-use",
      "SnapshotId": "snap-f23ec1c8",
      "CreateTime": "2013-09-17T00:55:03.000Z",
      "Size": 30
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-18T20:26:16.000Z",
          "InstanceId": "i-4b41a37c",
          "VolumeId": "vol-2e410a47",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-2e410a47",
      "State": "in-use",
      "SnapshotId": "snap-708e8348",
      "CreateTime": "2013-09-18T20:26:15.000Z",
      "Size": 8
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2020-11-20T19:54:06.000Z",
          "InstanceId": "i-1jd73kv8",
          "VolumeId": "vol-a1b3c7nd",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-a1b3c7nd",
      "State": "in-use",
      "SnapshotId": "snap-234087fb",
      "CreateTime": "2020-11-20T19:54:05.000Z",
      "Size": 15
    }
  ]
```

```
}
```

Identifiants

Les identificateurs sont les étiquettes des valeurs de sortie. Lorsque vous créez des filtres, vous utilisez des identifiants pour affiner les résultats de vos requêtes. Dans l'exemple de sortie suivant, tous les identificateurs tels que `VolumesAvailabilityZone`, et `AttachTime` sont surlignés.

```
$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-17T00:55:03.000Z",
          "InstanceId": "i-a071c394",
          "VolumeId": "vol-e11a5288",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-e11a5288",
      "State": "in-use",
      "SnapshotId": "snap-f23ec1c8",
      "CreateTime": "2013-09-17T00:55:03.000Z",
      "Size": 30
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-18T20:26:16.000Z",
          "InstanceId": "i-4b41a37c",
          "VolumeId": "vol-2e410a47",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
```

```
      "VolumeId": "vol-2e410a47",
      "State": "in-use",
      "SnapshotId": "snap-708e8348",
      "CreateTime": "2013-09-18T20:26:15.000Z",
      "Size": 8
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2020-11-20T19:54:06.000Z",
          "InstanceId": "i-1jd73kv8",
          "VolumeId": "vol-a1b3c7nd",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-a1b3c7nd",
      "State": "in-use",
      "SnapshotId": "snap-234087fb",
      "CreateTime": "2020-11-20T19:54:05.000Z",
      "Size": 15
    }
  ]
}
```

Pour plus d'informations, consultez la section Identifiants sur le JMESPath site Web.

Sélection dans une liste

Une liste ou un tableau est un identifiant suivi d'un crochet « `[` » tel que `Volumes` et `Attachments` dans lethe section called “Avant de commencer”.

Syntaxe

```
<listName>[ ]
```

Pour filtrer toutes les sorties d'un tableau, vous pouvez utiliser la notation générique. Les expressions génériques sont des expressions utilisées pour renvoyer des éléments à l'aide de la `*` notation.

L'exemple suivant interroge l'ensemble `Volumes` du contenu.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  }
]
```

Pour afficher un volume spécifique du tableau par index, vous appelez le tableau index. Par exemple, le premier élément du `Volumes` tableau a un indice de 0, ce qui donne lieu à la `Volumes[0]` requête. Pour plus d'informations sur les index de tableaux, consultez la section expressions d'index sur le JMESPathsite Web.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[0]'
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-e11a5288",
  "State": "in-use",
  "SnapshotId": "snap-f23ec1c8",
  "CreateTime": "2013-09-17T00:55:03.000Z",
  "Size": 30
}
```

Pour afficher une plage spécifique de volumes par index, utilisez `slice` la syntaxe suivante, où start est l'index du tableau de départ, stop est l'index où le filtre arrête le traitement et step est l'intervalle de saut.

Syntaxe

```
<arrayName>[<start>:<stop>:<step>]
```

Si l'une de ces valeurs est omise dans l'expression de la tranche, elle utilise les valeurs par défaut suivantes :

- Début — Le premier index de la liste, 0.
- Stop : dernier index de la liste.
- Étape — Aucune étape n'est sautée, la valeur étant 1.

Pour renvoyer uniquement les deux premiers volumes, vous devez utiliser une valeur de départ de 0, une valeur de fin de 2 et une valeur d'étape de 1, comme indiqué dans l'exemple suivant.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[0:2:1]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-18T20:26:16.000Z",
        "InstanceId": "i-4b41a37c",
        "VolumeId": "vol-2e410a47",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-2e410a47",
    "State": "in-use",
    "SnapshotId": "snap-708e8348",
    "CreateTime": "2013-09-18T20:26:15.000Z",
    "Size": 8
```

```
  }
]
```

Comme cet exemple contient des valeurs par défaut, vous pouvez raccourcir la tranche de `Volumes[0:2:1]` à`Volumes[:2]`.

L'exemple suivant omet les valeurs par défaut et renvoie tous les deux volumes de l'ensemble du tableau.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[::2]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
```

```
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  }
]
```

Les étapes peuvent également utiliser des nombres négatifs pour filtrer dans l'ordre inverse d'un tableau, comme illustré dans l'exemple suivant.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[::-2]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
```

```
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  }
]
```

Pour plus d'informations, consultez la section Slices sur le JMESPath site Web.

Filtrer les données imbriquées

Pour affiner le filtrage des valeurs imbriquées, vous pouvez utiliser des sous-expressions en ajoutant un point et vos critères de filtre. `Volumes[*]`

Syntaxe

```
<expression>.<expression>
```

L'exemple suivant montre toutes les `Attachments` informations pour tous les volumes.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments'
[
  [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
```

```
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ]
]
```

Pour filtrer davantage les valeurs imbriquées, ajoutez l'expression pour chaque identificateur imbriqué. L'exemple suivant répertorie les `State` pour tous`Volumes`.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

Aplatir les résultats

Pour plus d'informations, consultez SubExpressionsle JMESPathsite Web.

Vous pouvez aplatir les résultats `Volumes[*].Attachments[*].State` en supprimant la notation générique à l'origine de la `Volumes[*].Attachments[].State` requête. L'aplatissement fréquent est utile pour améliorer la lisibilité des résultats.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]
```

Pour plus d'informations, voir Flatten sur le JMESPath site Web.

Filtrage pour des valeurs spécifiques

Pour filtrer des valeurs spécifiques dans une liste, vous devez utiliser une expression de filtre, comme indiqué dans la syntaxe suivante.

Syntaxe

```
? <expression> <comparator> <expression>]
```

Les comparateurs d'expressions incluent ==!=,<,, <=>, et>=. L'exemple suivant filtre `VolumeIds` pour tous `Volumes` dans un `AttachedState`.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'
[
  [
    "vol-e11a5288"
  ],
  [
    "vol-2e410a47"
  ],
  [
    "vol-a1b3c7nd"
  ]
]
```

Cela peut ensuite être aplati, comme dans l'exemple suivant.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'
```

```
[
  "vol-e11a5288",
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

L'exemple suivant filtre tous ceux `VolumeIds` `Volumes` dont la taille est inférieure à 20.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?Size < `20`].VolumeId'
[
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

Pour plus d'informations, consultez Filtrer les expressions sur le JMESPathsite Web.

Expressions de tuyauterie

Vous pouvez rediriger les résultats d'un filtre vers une nouvelle liste, puis filtrer le résultat avec une autre expression en utilisant la syntaxe suivante :

Syntaxe

```
<expression> | <expression>]
```

L'exemple suivant prend les résultats du filtre de l'`Volumes[*].Attachments[].InstanceId`expression et affiche le premier résultat dans le tableau.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].InstanceId | [0]'
"i-a071c394"
```

Pour ce faire, cet exemple crée d'abord le tableau à partir de l'expression suivante.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].InstanceId'
"i-a071c394",
  "i-4b41a37c",
```

```
  "i-1jd73kv8"
```

Puis renvoie le premier élément de ce tableau.

```
"i-a071c394"
```

Pour plus d'informations, consultez la section Pipe Expressions sur le JMESPathsite Web.

Filtrage pour plusieurs valeurs d'identifiant

Pour filtrer plusieurs identificateurs, vous devez utiliser une liste à sélection multiple en utilisant la syntaxe suivante :

Syntaxe

```
<listName>[].[<expression>, <expression>]
```

Dans l'exemple suivant, `VolumeId` et `VolumeType` sont filtrés dans la `Volumes` liste, ce qui donne l'expression suivante.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].[VolumeId, VolumeType]'
[
  [
    "vol-e11a5288",
    "standard"
  ],
  [
    "vol-2e410a47",
    "standard"
  ],
  [
    "vol-a1b3c7nd",
    "standard"
  ]
]
```

Pour ajouter des données imbriquées à la liste, vous devez ajouter une autre liste de sélection multiple. L'exemple suivant développe l'exemple précédent en filtrant également pour `InstanceId` et `State` dans la `Attachments` liste imbriquée. Cela donne l'expression suivante.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].[VolumeId, VolumeType, Attachments[].[InstanceId, State]]'
[
  [
    "vol-e11a5288",
    "standard",
    [
      [
        "i-a071c394",
        "attached"
      ]
    ]
  ],
  [
    "vol-2e410a47",
    "standard",
    [
      [
        "i-4b41a37c",
        "attached"
      ]
    ]
  ],
  [
    "vol-a1b3c7nd",
    "standard",
    [
      [
        "i-1jd73kv8",
        "attached"
      ]
    ]
  ]
]
```

Pour qu'elle soit plus lisible, aplatissez l'expression comme indiqué dans l'exemple suivant.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].[VolumeId, VolumeType, Attachments[].[InstanceId, State][]][]'
[
  "vol-e11a5288",
  "standard",
  [
```

```
    "i-a071c394",
    "attached"
  ],
  "vol-2e410a47",
  "standard",
  [
    "i-4b41a37c",
    "attached"
  ],
  "vol-a1b3c7nd",
  "standard",
  [
    "i-1jd73kv8",
    "attached"
  ]
]
```

Pour plus d'informations, consultez la section Liste de sélections multiples sur le JMESPath site Web.

Ajouter des étiquettes aux valeurs d'identification

Pour faciliter la lecture de cette sortie, utilisez un hachage à sélection multiple avec la syntaxe suivante.

Syntaxe

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

Il n'est pas nécessaire que votre étiquette d'identification soit identique au nom de l'identifiant. L'exemple suivant utilise l'étiquette `VolumeType` pour les `VolumeType` valeurs.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].{VolumeType: VolumeType}'
[
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
```

```
  }
]
```

Pour des raisons de simplicité, l'exemple suivant conserve les noms d'identifiant pour chaque étiquette et affiche les noms `VolumeIdVolumeType`,`InstanceId`, et `State` pour tous les volumes :

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].{VolumeId: VolumeId, VolumeType: VolumeType, InstanceId:
 Attachments[0].InstanceId, State: Attachments[0].State}'
[
  {
    "VolumeId": "vol-e11a5288",
    "VolumeType": "standard",
    "InstanceId": "i-a071c394",
    "State": "attached"
  },
  {
    "VolumeId": "vol-2e410a47",
    "VolumeType": "standard",
    "InstanceId": "i-4b41a37c",
    "State": "attached"
  },
  {
    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
  }
]
```

Pour plus d'informations, consultez la section Multiselect Hash sur le JMESPath site Web.

Fonctions

La JMESPath syntaxe contient de nombreuses fonctions que vous pouvez utiliser pour vos requêtes. Pour plus d'informations sur JMESPath les fonctions, voir Fonctions intégrées sur le JMESPathsite Web.

Pour montrer comment intégrer une fonction dans vos requêtes, l'exemple suivant utilise cette `sort_by` fonction. La `sort_by` fonction trie un tableau en utilisant une expression comme clé de tri selon la syntaxe suivante :

Syntaxe

```
sort_by(<listName>, <sort expression>)[].<expression>
```

L'exemple suivant utilise l'exemple de hachage à sélection multiple précédent et trie la sortie par. `VolumeId`

```
$ aws ec2 describe-volumes \
    --query 'sort_by(Volumes, &VolumeId)[].{VolumeId: VolumeId, VolumeType: VolumeType,
 InstanceId: Attachments[0].InstanceId, State: Attachments[0].State}'
[
  {
    "VolumeId": "vol-2e410a47",
    "VolumeType": "standard",
    "InstanceId": "i-4b41a37c",
    "State": "attached"
  },
  {
    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
  },
  {
    "VolumeId": "vol-e11a5288",
    "VolumeType": "standard",
    "InstanceId": "i-a071c394",
    "State": "attached"
  }
]
```

Pour plus d'informations, consultez sort_by sur le site Web. JMESPath

`--query`Exemples avancés

Pour extraire des informations d'un élément spécifique

L'exemple suivant utilise le paramètre `--query` pour rechercher un élément spécifique dans une liste, puis extrait les informations de cet élément. L'exemple répertorie tous les éléments `AvailabilityZones` associés au point de terminaison de service spécifié. Il extrait l'élément de la liste `ServiceDetails` pour lequel `ServiceName` est spécifié, puis il génère en sortie le champ `AvailabilityZones` à partir de cet élément sélectionné.

```
$ aws --region us-east-1 ec2 describe-vpc-endpoint-services \
    --query 'ServiceDetails[?ServiceName==`com.amazonaws.us-
east-1.ecs`].AvailabilityZones'
[
    [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ]
]
```

Pour afficher les instantanés après la date de création spécifiée

L'exemple suivant montre comment référencer tous vos instantanés qui ont été créés après une date spécifiée, y compris certains champs disponibles dans la sortie.

```
$ aws ec2 describe-snapshots --owner self \
    --output json \
    --query 'Snapshots[?StartTime>=`2018-02-07`].
{Id:SnapshotId,VId:VolumeId,Size:VolumeSize}'
[
    {
        "id": "snap-0effb42b7a1b2c3d4",
        "vid": "vol-0be9bb0bf12345678",
        "Size": 8
    }
]
```

Pour afficher le plus récent AMIs

L'exemple suivant répertorie les cinq images Amazon Machine (AMIs) les plus récentes que vous avez créées, triées de la plus récente à la plus ancienne.

```
$ aws ec2 describe-images \
    --owners self \
    --query 'reverse(sort_by(Images,&CreationDate))[:5].{id:ImageId,date:CreationDate}'
[
    {
        "id": "ami-0a1b2c3d4e5f60001",
```

```
        "date": "2018-11-28T17:16:38.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60002",
        "date": "2018-09-15T13:51:22.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60003",
        "date": "2018-08-19T10:22:45.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60004",
        "date": "2018-05-03T12:04:02.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60005",
        "date": "2017-12-13T17:16:38.000Z"
    }
]
```

Pour afficher les instances Auto Scaling défectueuses

L'exemple suivant montre uniquement `InstanceId` les instances défectueuses du groupe Auto Scaling spécifié.

```
$ aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-name My-AutoScaling-Group-Name \
    --output text \
    --query 'AutoScalingGroups[*].Instances[?HealthStatus==`Unhealthy`].InstanceId'
```

Pour inclure des volumes dotés de la balise spécifiée

L'exemple suivant décrit toutes les instances dotées d'une `test` balise. Tant qu'une autre balise est `test` attachée au volume à côté, le volume est toujours renvoyé dans les résultats.

Expression ci-dessous pour renvoyer toutes les balises dont la `test` balise est dans un tableau. Toutes les balises qui ne sont pas les `test` balises contiennent une `null` valeur.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Tags[?Value == `test`]'
```

Pour exclure les volumes dotés de la balise spécifiée

L'exemple suivant décrit toutes les instances sans `test` balise. L'utilisation d'une `?Value != `test`` expression simple ne fonctionne pas pour exclure un volume, car les volumes peuvent avoir plusieurs balises. Tant qu'une autre balise est `test` attachée au volume à côté, le volume est toujours renvoyé dans les résultats.

Pour exclure tous les volumes dotés de `test` cette balise, commencez par l'expression ci-dessous pour renvoyer toutes les balises contenant la `test` balise dans un tableau. Toutes les balises qui ne sont pas les `test` balises contiennent une `null` valeur.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Tags[?Value == `test`]'
```

Filtrez ensuite tous les `test` résultats positifs à l'aide de la `not_null` fonction.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?!not_null(Tags[?Value == `test`].Value)]'
```

Répartissez les résultats pour les aplatir, ce qui donne lieu à la requête suivante.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?!not_null(Tags[?Value == `test`].Value)] | []'
```

Combinaison du filtrage côté serveur et côté client

Vous pouvez utiliser conjointement le filtrage côté serveur et côté client. Le filtrage côté serveur est effectué en premier, ce qui envoie les données au client que le `--query` paramètre filtre ensuite. Si vous utilisez de grands ensembles de données, le fait d'utiliser d'abord le filtrage côté serveur peut réduire la quantité de données envoyées au client pour chaque AWS CLI appel, tout en conservant la puissante personnalisation qu'offre le filtrage côté client.

L'exemple suivant répertorie les EC2 volumes Amazon utilisant le filtrage côté serveur et côté client. Le service filtre une liste de tous les volumes attachés dans la zone de `us-west-2a` disponibilité. Le paramètre `--query` supplémentaire limite la sortie uniquement vers les volumes avec une valeur `Size` supérieure à 50, et indique uniquement les champs spécifiés avec des noms définis par l'utilisateur.

```
$ aws ec2 describe-volumes \
    --filters "Name=availability-zone,Values=us-west-2a" "Name=status,Values=attached"
 \
```

```
    --query 'Volumes[?Size > `50`].{Id:VolumeId,Size:Size,Type:VolumeType}'
[
    {
        "Id": "vol-0be9bb0bf12345678",
        "Size": 80,
        "VolumeType": "gp2"
    }
]
```

L'exemple suivant récupère une liste d'images qui répondent à plusieurs critères. Il utilise ensuite le paramètre `--query` pour trier la sortie `CreationDate`, en sélectionnant uniquement les plus récentes. Il affiche ensuite le `ImageId` de cette seule image.

```
$ aws ec2 describe-images \
    --owners amazon \
    --filters "Name=name,Values=amzn*gp2" "Name=virtualization-type,Values=hvm"
 "Name=root-device-type,Values=ebs" \
    --query "sort_by(Images, &CreationDate)[-1].ImageId" \
    --output text
ami-00ced3122871a4921
```

L'exemple suivant montre comment afficher le nombre de volumes disponibles qui sont supérieurs à 1 000 E/S par seconde en utilisant `length` pour compter leur nombre dans une liste.

```
$ aws ec2 describe-volumes \
    --filters "Name=status,Values=available" \
    --query 'length(Volumes[?Iops > `1000`])'
3
```

L'exemple suivant récupère les noms des groupes Auto Scaling qui utilisent les configurations de lancement spécifiées Région AWS qui utilisent des AWS CloudFormation stacks.

```
$ aws autoscaling describe-auto-scaling-groups --region us-west-2 \
  --filters Name=tag-key,Values=aws:cloudformation:stack-name \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`].AutoScalingGroupName'
[
    "group-1",
    "group-2",
    "group-3"
]
```

Ressources supplémentaires

AWS CLI invite automatique

Lorsque vous commencez à utiliser des expressions de filtre, vous pouvez utiliser la fonction d'invite automatique de la AWS CLI version 2. La fonction d'invite automatique fournit un aperçu lorsque vous appuyez sur la touche F5. Pour de plus amples informations, veuillez consulter the section called “Invite automatique”.

JMESPath Borne

JMESPath Terminal est une commande de terminal interactive permettant d'expérimenter JMESPath des expressions utilisées pour le filtrage côté client. À l'aide de `jpterm` cette commande, le terminal affiche les résultats de la requête immédiatement au fur et à mesure que vous tapez. Vous pouvez directement diriger la AWS CLI sortie vers le terminal, ce qui permet d'expérimenter des requêtes avancées.

L'exemple suivant indique des canaux `aws ec2 describe-volumes` de sortie directement vers JMESPath le terminal.

```
$ aws ec2 describe-volumes | jpterm
```

Pour plus d'informations sur le JMESPath terminal et les instructions d'installation, voir JMESPathTerminal activé GitHub.

utilitaire jq

L'`jq`utilitaire vous permet de transformer votre sortie côté client dans le format de sortie que vous souhaitez. Pour plus d'informations `jq` et des instructions d'installation, consultez jq on GitHub.

Codes de retour de ligne de commande dans le AWS CLI

Le code de retour est généralement un code caché envoyé après l'exécution d'une commande AWS Command Line Interface (AWS CLI) qui décrit l'état de la commande. Vous pouvez utiliser la `echo` commande pour afficher le code envoyé depuis la dernière AWS CLI commande et utiliser ces codes pour déterminer si une commande a réussi ou si elle a échoué, et pourquoi une commande peut contenir une erreur. Outre les codes de retour, vous pouvez consulter plus de détails sur une panne en exécutant vos commandes avec le `--debug` commutateur. Ce commutateur produit un rapport détaillé des étapes AWS CLI utilisées pour traiter la commande et du résultat de chaque étape.

Pour déterminer le code de retour d'une AWS CLI commande, exécutez l'une des commandes suivantes immédiatement après avoir exécuté la commande CLI.

Linux and macOS

```
$ echo $?
0
```

Windows PowerShell

```
PS> echo $lastexitcode
0
```

Windows Command Prompt

```
C:\> echo %errorlevel%
0
```

Les valeurs de code de retour qui peuvent être renvoyées à la fin de l'exécution d'une commande AWS Command Line Interface (AWS CLI) sont les suivantes.

Code	Signification
0	Le service a répondu avec un code d'état de réponse HTTP de 200 indiquant qu'aucune erreur n'avait été générée par le AWS service AWS CLI et auquel la demande avait été envoyée.
1	Une ou plusieurs opérations de transfert Amazon S3 ont échoué. Limité aux commandes S3.
2	La signification de ce code de retour dépend de la commande. • Applicable à toutes les AWS CLI commandes : la commande saisie n'a pas pu être analysée. Les erreurs d'analyse peuvent être dus, notamment, à des sous-commandes ou des arguments obligatoires manquants ou à l'utilisation de commandes ou d'argumen ts inconnus. • Limité aux commandes S3 : un ou plusieurs fichiers marqués pour le transfert ont été ignorés pendant le processus de transfert. En revanche, tous les autres fichiers marqués

Code	Signification
	pour transfert ont été correctement transférés. Les fichiers qui sont ignorés au cours du processus de transfert sont les suivants : fichiers inexistants, fichiers spéciaux de périphérique en mode caractère et en mode bloc, files d'attente FIFO ou sockets et fichiers pour lesquels l'utilisateur ne dispose pas d'autorisations de lecture.
130	La commande a été interrompue par un SIGINT. Il s'agit du signal que vous envoyez pour annuler une commande avec le signe `Ctrl +C`.
252	La syntaxe de la commande n'était pas valide, un paramètre inconnu a été fourni ou une valeur de paramètre incorrecte a empêché l'exécution de la commande.
253	L'environnement ou la configuration du système n'était pas valide. Bien que la commande fournie soit syntaxiquement valide, l'absence de configuration ou d'informations d'identif ication a empêché l'exécution de la commande.
254	La commande a été correctement analysée et une demande a été envoyée au service spécifié, mais le service a renvoyé une erreur. Cela indique généralement une utilisation incorrecte de l'API ou d'autres problèmes spécifiques au service.
255	La commande a échoué. Des erreurs ont été générées par le AWS CLI ou par le AWS service auquel la demande a été envoyée.

Utilisation d'assistants personnalisés pour exécuter des commandes interactives dans AWS CLI

Le AWS Command Line Interface (AWS CLI) permet d'utiliser un assistant pour certaines commandes. Pour contribuer ou consulter la liste complète des AWS CLI assistants disponibles, consultez le dossier des AWS CLI assistants sur. GitHub

Comment ça marche

Semblable à la AWS console, AWS CLI elle dispose d'un assistant d'interface utilisateur qui vous guide dans la gestion de vos AWS ressources. Pour utiliser l'assistant, vous devez appeler la `wizard` sous-commande et le nom de l'assistant après le nom du service dans une commande. La structure de commande est la suivante :

Syntaxe :

```
$ aws <command> wizard <wizardName>
```

L'exemple suivant appelle l'assistant pour créer une nouvelle dynamodb table.

```
$ aws dynamodb wizard new-table
```

`aws configure`est le seul magicien qui n'a pas de nom de magicien. Lorsque vous exécutez l'assistant, exécutez la `aws configure wizard` commande comme le montre l'exemple suivant :

```
$ aws configure wizard
```

Après avoir appelé un assistant, un formulaire s'affiche dans le shell. Pour chaque paramètre, une liste d'options vous est fournie ou vous êtes invité à saisir une chaîne. Pour effectuer une sélection dans une liste, utilisez les flèches haut et bas et appuyez sur ENTER. Pour afficher les détails d'une option, appuyez sur la flèche droite. Lorsque vous avez terminé de renseigner un paramètre, appuyez sur ENTER.

```
$ aws configure wizard
What would you like to configure
> Static Credentials
  Assume Role
  Process Provider
  Additional CLI configuration
Enter the name of the profile:
Enter your Access Key Id:
Enter your Secret Access Key:
```

Pour modifier les instructions précédentes, utilisez SHIFT + TAB. Pour certains assistants, après avoir rempli toutes les instructions, vous pouvez prévisualiser un AWS CloudFormation modèle ou la AWS CLI commande contenant vos informations. Ce mode de prévisualisation est utile pour apprendre à utiliser AWS CLI, à gérer APIs et à créer des modèles pour les scripts.

Appuyez sur ENTER après avoir prévisualisé ou à la dernière invite pour exécuter la commande finale.

```
$ aws configure wizard
```

```
What would you like to configure
Enter the name of the profile: testWizard
Enter your Access Key Id: AB1C2D3EF4GH5I678J90K
Enter your Secret Access Key: ab1c2def34gh5i67j8k90l1mnop2qr3s45tu678v90
<ENTER>
```

Création et utilisation d'alias dans AWS CLI

Les alias sont des raccourcis que vous pouvez créer dans le AWS Command Line Interface (AWS CLI) pour raccourcir les commandes ou les scripts que vous utilisez fréquemment. Vous créez des alias dans le `alias` fichier situé dans votre dossier de configuration.

Rubriques

- Prérequis
- Étape 1 : Création du fichier d'alias
- Étape 2 : Création d'un alias
- Étape 3 : appeler un alias
- Exemples de référentiels d'alias
- Ressources

Prérequis

Pour utiliser les commandes d'alias, vous devez effectuer les opérations suivantes :

- Installation et configuration de l' AWS CLI. Pour plus d’informations, consultez the section called “Installation/mise à jour” et Authentification et informations d'accès pour AWS CLI.
- Utilisez une AWS CLI version minimale de 1.11.24 ou 2.0.0.
- (Facultatif) Pour utiliser des scripts AWS CLI alias bash, vous devez utiliser un terminal compatible avec bash.

Étape 1 : Création du fichier d'alias

Pour créer le `alias` fichier, vous pouvez utiliser la navigation dans les fichiers et un éditeur de texte, ou utiliser votre terminal préféré en suivant la step-by-step procédure. Pour créer rapidement votre fichier d'alias, utilisez le bloc de commande suivant.

Linux and macOS

```
$ mkdir -p ~/.aws/cli
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> md %USERPROFILE%\.aws\cli
C:\> echo [toplevel] > %USERPROFILE%/.aws/cli/alias
```

Pour créer le fichier d'alias

1. Créez un dossier nommé `cli` dans votre dossier AWS CLI de configuration. Par défaut, le dossier de configuration `~/.aws/` se trouve sous Linux ou macOS et `%USERPROFILE%\.aws\` sous Windows. Vous pouvez le créer via votre navigation dans les fichiers ou à l'aide de la commande suivante.

   Linux and macOS

   ```
   $ mkdir -p ~/.aws/cli
   ```

   Windows

   ```
   C:\> md %USERPROFILE%\.aws\cli
   ```

   Le chemin par défaut du `cli` dossier obtenu est `~/.aws/cli/` sous Linux ou macOS et `%USERPROFILE%\.aws\cli` sous Windows.

2. Dans le `cli` dossier, créez un fichier texte `alias` sans extension et ajoutez-le `[toplevel]` à la première ligne. Vous pouvez créer ce fichier via votre éditeur de texte préféré ou utiliser la commande suivante.

   Linux and macOS

   ```
   $ echo '[toplevel]' > ~/.aws/cli/alias
   ```

Windows

```
C:\> echo [toplevel] > %USERPROFILE%/.aws/cli/alias
```

Étape 2 : Création d'un alias

Vous pouvez créer un alias à l'aide de commandes de base ou de scripts bash.

Création d'un alias de commande de base

Vous pouvez créer votre alias en ajoutant une commande en utilisant la syntaxe suivante dans le `alias` fichier que vous avez créé à l'étape précédente.

Syntaxe

```
aliasname = command [--options]
```

*`aliasname`*C'est ce que tu appelles ton alias. *`command`*Il s'agit de la commande que vous souhaitez appeler, qui peut inclure d'autres alias. Vous pouvez inclure des options ou des paramètres dans votre alias, ou les ajouter lorsque vous appelez votre alias.

L'exemple suivant crée un alias nommé à `aws whoami` l'aide de la commande. Comme cet alias appelle une AWS CLI commande existante, vous pouvez écrire la commande sans le `aws` préfixe.

```
whoami = sts get-caller-identity
```

L'exemple suivant reprend l'`whoami`exemple précédent et ajoute les `output` options de `Account` filtre et de texte.

```
whoami2 = sts get-caller-identity --query Account --output text
```

Création d'un alias de sous-commande

> **Note**
>
> La fonctionnalité d'alias de sous-commande nécessite une AWS CLI version minimale de 1.11.24 ou 2.0.0

Vous pouvez créer un alias pour les sous-commandes en ajoutant une commande à l'aide de la syntaxe suivante dans le `alias` fichier que vous avez créé à l'étape précédente.

Syntaxe

```
[command commandGroup]
aliasname = command [--options]
```

*commandGroup*Il s'agit de l'espace de noms de commande, par exemple la commande `aws ec2 describe-regions` se trouve sous le groupe de `ec2` commandes. *aliasname*C'est ce que tu appelles ton alias. *command*Il s'agit de la commande que vous souhaitez appeler, qui peut inclure d'autres alias. Vous pouvez inclure des options ou des paramètres dans votre alias, ou les ajouter lorsque vous appelez votre alias.

L'exemple suivant crée un alias nommé à `aws ec2 regions` l'aide de la commande. Comme cet alias appelle une AWS CLI commande existante sous l'espace de noms de `ec2` commande, vous pouvez écrire la commande sans le `aws ec2` préfixe.

```
[command ec2]
regions = describe-regions --query Regions[].RegionName
```

Pour créer des alias à partir de commandes situées en dehors de l'espace de noms des commandes, préfixez la commande complète d'un point d'exclamation. L'exemple suivant crée un alias nommé à `aws ec2 instance-profiles` l'aide de la commande.

```
[command ec2]
instance-profiles = !aws iam list-instance-profiles
```

> **Note**
>
> Les alias utilisent uniquement les espaces de noms de commande existants et vous ne pouvez pas en créer de nouveaux. Par exemple, vous ne pouvez pas créer d'alias avec la `[command johnsmith]` section car l'espace de noms de `johnsmith` commande n'existe pas déjà.

Création d'un alias de script bash

> **Warning**
>
> Pour utiliser des scripts AWS CLI alias bash, vous devez utiliser un terminal compatible avec bash

Vous pouvez créer un alias à l'aide de scripts bash pour des processus plus avancés en utilisant la syntaxe suivante.

Syntaxe

```
aliasname =
    !f() {
        script content
}; f
```

*aliasname*C'est ce que vous appelez votre alias et *script content* c'est le script que vous souhaitez exécuter lorsque vous appelez l'alias.

L'exemple suivant permet `opendns` de générer votre adresse IP actuelle. Comme vous pouvez utiliser des alias dans d'autres alias, l'`myip`alias suivant est utile pour autoriser ou révoquer l'accès à votre adresse IP depuis d'autres alias.

```
myip =
  !f() {
    dig +short myip.opendns.com @resolver1.opendns.com
  }; f
```

L'exemple de script suivant appelle l'`aws myip`alias précédent pour autoriser votre adresse IP à accéder à un groupe EC2 de sécurité Amazon.

```
authorize-my-ip =
  !f() {
    ip=$(aws myip)
    aws ec2 authorize-security-group-ingress --group-id ${1} --cidr $ip/32 --protocol
 tcp --port 22
  }; f
```

Lorsque vous appelez des alias utilisant des scripts bash, les variables sont toujours transmises dans l'ordre dans lequel vous les avez saisies. Dans les scripts bash, les noms des variables ne sont pas pris en compte, mais uniquement l'ordre dans lequel elles apparaissent. Dans l'exemple d'`textalert`alias suivant, la variable de l'`--message`option est la première et `--phone-number` l'option la deuxième.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

Étape 3 : appeler un alias

Pour exécuter l'alias que vous avez créé dans votre `alias` fichier, utilisez la syntaxe suivante. Vous pouvez ajouter des options supplémentaires lorsque vous appelez votre alias.

Syntaxe

```
$ aws aliasname
```

L'exemple suivant utilise l'alias de `aws whoami` commande.

```
$ aws
 whoami
{
    "UserId": "A12BCD34E5FGHI6JKLM",
    "Account": "1234567890987",
    "Arn": "arn:aws:iam::1234567890987:user/userName"
}
```

L'exemple suivant utilise l'`aws whoami`alias avec des options supplémentaires pour renvoyer uniquement le `Account` nombre en `text` sortie.

```
$ aws whoami --query Account --output
 text
1234567890987
```

L'exemple suivant utilise l'alias de `aws ec2 regions` sous-commande.

```
$ aws ec2
 regions
[
    "ap-south-1",
    "eu-north-1",
    "eu-west-3",
    "eu-west-2",
...
```

Appel d'un alias à l'aide de variables de script bash

Lorsque vous appelez des alias utilisant des scripts bash, les variables sont transmises dans l'ordre dans lequel elles ont été saisies. Dans les scripts bash, le nom des variables n'est pas pris en compte, mais uniquement l'ordre dans lequel elles apparaissent. Par exemple, dans l'`textalert`alias suivant, la variable de l'option `--message` est première et `--phone-number` seconde.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

Lorsque vous appelez l'`textalert`alias, vous devez transmettre les variables dans le même ordre que celui dans lequel elles sont exécutées dans l'alias. Dans l'exemple suivant, nous utilisons les variables `$message` et`$phone`. La `$message` variable est passée comme `${1}` pour l'`--message`option et la `$phone` variable est passée comme `${2}` pour l'`--phone-number`option. Cela permet d'appeler correctement l'`textalert`alias pour envoyer un message.

```
$ aws textalert $message
 $phone
{
    "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

Dans l'exemple suivant, l'ordre est changé lorsque l'alias est appelé vers `$phone` et`$message`. La `$phone` variable est passée comme `${1}` pour l'`--message`option et la `$message` variable est passée comme `${2}` pour l'`--phone-number`option. Comme les variables ne sont pas ordonnées, l'alias les transmet de manière incorrecte. Cela provoque une erreur car le contenu de `$message`

ne correspond pas aux exigences de formatage des numéros de téléphone pour l'`--phone-number`option.

```
$ aws textalert $phone
 $message
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help

Unknown options: text
```

Exemples de référentiels d'alias

Le référentiel d'AWS CLI alias GitHubcontient des exemples d' AWS CLI alias créés par l'équipe de AWS CLI développeurs et la communauté. Vous pouvez utiliser l'exemple de `alias` fichier complet ou utiliser des alias individuels pour votre propre usage.

> **Warning**
>
> L'exécution des commandes de cette section supprime votre `alias` fichier existant. Pour éviter de remplacer votre fichier d'alias existant, modifiez l'emplacement de votre téléchargement.

Pour utiliser les alias du référentiel

1. Installez Git. Pour les instructions d'installation, consultez Getting Started - Install Git dans la documentation Git.
2. Installez la `jp` commande. La `jp` commande est utilisée dans l'`tostring`alias. Pour les instructions d'installation, consultez le fichier JMESPath (jp) README.md sur. GitHub
3. Installez la `jq` commande. La `jq` commande est utilisée dans l'`tostring-with-jq`alias. Pour les instructions d'installation, consultez le processeur JSON (jq) activé GitHub.
4. Téléchargez le `alias` fichier en effectuant l'une des opérations suivantes :

   - Exécutez les commandes suivantes pour télécharger le `alias` fichier depuis le référentiel et le copier dans votre dossier de configuration.

Linux and macOS

```
$ git clone https://github.com/awslabs/awscli-aliases.git
$ mkdir -p ~/.aws/cli
$ cp awscli-aliases/alias ~/.aws/cli/alias
```

Windows

```
C:\> git clone https://github.com/awslabs/awscli-aliases.git
C:\> md %USERPROFILE%\.aws\cli
C:\> copy awscli-aliases\alias %USERPROFILE%\.aws\cli
```

- Téléchargez-le directement depuis le référentiel et enregistrez-le dans le `cli` dossier de votre dossier AWS CLI de configuration. Par défaut, le dossier de configuration `~/.aws/` se trouve sous Linux ou macOS et `%USERPROFILE%\.aws\` sous Windows.

5. Pour vérifier que les alias fonctionnent, exécutez l'alias suivant.

```
$ aws whoami
```

Cela affiche la même réponse que la `aws sts get-caller-identity` commande :

```
{
    "Account": "012345678901",
    "UserId": "AIUAINBADX2VEG2TC6HD6",
    "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```

Ressources

- Le référentiel d'AWS CLI alias GitHubcontient des exemples d' AWS CLI alias créés par l'équipe de AWS CLI développeurs et la contribution de la AWS CLI communauté.
- L'annonce de la fonctionnalité d'alias publiée dans AWS re:Invent 2016 : The Effective AWS CLI User on. YouTube
- `aws sts get-caller-identity`
- `aws ec2 describe-instances`
- `aws sns publish`

Résolution des erreurs liées au AWS CLI

Cette section décrit les erreurs courantes et les étapes de résolution des problèmes à suivre pour résoudre votre problème. Nous vous suggérons de suivre d'abord le dépannage général.

Table des matières

- Résolution des problèmes généraux à essayer en premier
  - Vérifiez le formatage AWS CLI de vos commandes
  - Vérifiez que Région AWS votre AWS CLI commande utilise
  - Vérifiez que vous utilisez une version récente du AWS CLI
  - Utilisez l'--debugoption
  - Activer et consulter les journaux d'historique des AWS CLI commandes
  - Confirmez AWS CLI que votre
- Erreurs de commande non trouvées
- La commande « aws --version » renvoie une version différente de celle que vous avez installée
- La commande « aws --version » renvoie une version après la désinstallation du AWS CLI
- Le a AWS CLI traité une commande avec un nom de paramètre incomplet
- Erreurs d'accès refusé
- Informations d'identification non valides et erreurs clés
- La signature ne correspond pas aux erreurs
- Erreurs de certificat SSL
- Erreurs JSON non valides
- Ressources supplémentaires

Résolution des problèmes généraux à essayer en premier

Si vous recevez un message d'erreur ou rencontrez un problème avec le AWS CLI, nous vous suggérons de suivre les conseils généraux suivants pour vous aider à résoudre le problème.

Retour en haut de page

Vérifiez le formatage AWS CLI de vos commandes

Si vous recevez un message d'erreur indiquant qu'une commande n'existe pas ou qu'elle ne reconnaît pas un paramètre (`Parameter validation failed`) qui, selon la documentation, est disponible, votre commande est peut-être mal formatée. Nous vous conseillons de vérifier les points suivants :

- Vérifiez que votre commande ne comporte pas d'erreurs d'orthographe ou de mise en forme.
- Vérifiez que tous les guillemets et les échappements appropriés à votre terminal sont corrects dans votre commande.
- Générez un AWS CLI squelette pour confirmer votre structure de commande.
- Pour le JSON, consultez le dépannage supplémentaire pour les valeurs JSON. Si vous rencontrez des problèmes avec le traitement du formatage JSON par votre terminal, nous vous suggérons d'ignorer les règles de citation du terminal en utilisant des blobs pour transmettre les données JSON directement au. AWS CLI

Pour plus d'informations sur la manière dont une commande spécifique doit être structurée, consultez le version 2 du guide de référence.

Retour en haut de page

Vérifiez que Région AWS votre AWS CLI commande utilise

> **Note**
>
> Vous devez spécifier un Région AWS lorsque vous utilisez le AWS CLI, soit explicitement, soit en définissant une région par défaut. Pour obtenir la liste de tous les éléments Régions AWS que vous pouvez spécifier, consultez la section AWS Régions et points de terminaison dans le Référence générale d'Amazon Web Services. Les Région AWS désignateurs utilisés AWS CLI sont les mêmes que ceux que vous voyez dans les points de terminaison AWS Management Console URLs de service.

Des erreurs ou des résultats inattendus peuvent se produire si un Service AWS n'est pas disponible pour ce que vous avez spécifié Région AWS ou si vos ressources se trouvent dans un autre endroit Région AWS. Par ordre de priorité, le Région AWS est défini de la manière suivante :

- L'option `--region` de ligne de commande.

- Variable d'`AWS_REGION`environnement compatible avec le SDK.
- La variable d'`AWS_DEFAULT_REGION`environnement.
- Le réglage du `region`profil.

Vérifiez que vous utilisez Région AWS les bonnes ressources.

Retour en haut de page

Vérifiez que vous utilisez une version récente du AWS CLI

Si vous recevez un message d'erreur indiquant qu'une commande n'existe pas ou qu'elle ne reconnaît pas un paramètre indiqué comme disponible dans le de référence AWS CLI version 2, vérifiez d'abord que votre commande est correctement formatée. Si le formatage est correct, nous vous recommandons de passer à la version la plus récente du AWS CLI. Des versions mises à jour AWS CLI sont publiées presque tous les jours ouvrables. De nouveaux AWS services, fonctionnalités et paramètres sont introduits dans ces nouvelles versions du AWS CLI. Le seul moyen d'accéder à ces nouveaux services, fonctions ou paramètres consiste à effectuer une mise à niveau vers une version publiée après la première introduction de cet élément.

La façon dont vous mettez à jour votre version du AWS CLI dépend de la manière dont vous l'avez initialement installée, comme décrit dans the section called “Installation/mise à jour”.

Si vous avez utilisé l'un des programmes d'installation fournis, vous devrez peut-être supprimer l'installation existante avant de télécharger et d'installer la dernière version de votre système d'exploitation.

Retour en haut de page

Utilisez l'**`--debug`**option

Lorsque le AWS CLI signale une erreur que vous ne comprenez pas immédiatement ou produit des résultats auxquels vous ne vous attendez pas, vous pouvez obtenir plus de détails sur l'erreur en exécutant à nouveau la commande avec l'`--debug`option. Avec cette option, les AWS CLI résultats fournissent des informations détaillées sur chaque étape nécessaire au traitement de votre commande. Les détails de la sortie peuvent vous aider à déterminer à quel moment l'erreur se produit et fournissent des indices sur son origine.

Vous pouvez envoyer le résultat dans un fichier texte pour une révision ultérieure, ou pour l'envoyer AWS Support lorsqu'on vous le demande.

Lorsque vous incluez l'`--debug`option, certains des détails incluent :

- La recherche d'informations d'identification
- L'analyse des paramètres fournis
- Construction de la requête envoyée aux AWS serveurs
- Le contenu de la demande envoyée à AWS
- Le contenu de la réponse brute
- La sortie formatée

Voici un exemple de commande exécutée avec et sans l'option `--debug` :

```
$ aws iam list-groups --profile MyTestProfile
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "MyTestGroup",
            "GroupId": "AGPA0123456789EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
            "CreateDate": "2019-08-12T19:34:04Z"
        }
    ]
}
```

```
$ aws iam list-groups --profile MyTestProfile --debug
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.16.215 Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - Arguments entered to
 CLI: ['iam', 'list-groups', '--debug']
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function add_scalar_parsers at 0x7fdf173161e0>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function register_uri_param_handler at 0x7fdf17dec400>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function inject_assume_role_provider_cache at
 0x7fdf17da9378>
2019-08-12 12:36:18,307 - MainThread - botocore.credentials - DEBUG - Skipping
 environment variable credential check because profile name was explicitly set.
2019-08-12 12:36:18,307 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function attach_history_handler at 0x7fdf173ed9d8>
```

```
2019-08-12 12:36:18,308 - MainThread - botocore.loaders - DEBUG - Loading JSON
 file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
service-2.json
2019-08-12 12:36:18,317 - MainThread - botocore.hooks - DEBUG - Event building-command-
table.iam: calling handler <function add_waiters at 0x7fdf1731a840>
2019-08-12 12:36:18,320 - MainThread - botocore.loaders - DEBUG - Loading JSON
 file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
waiters-2.json
2019-08-12 12:36:18,321 - MainThread - awscli.clidriver - DEBUG - OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
 <awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
 <awscli.arguments.CLIArgument object at 0x7fdf171b09b0>)])
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_streaming_output_arg at
 0x7fdf17316510>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_cli_input_json at
 0x7fdf17da9d90>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function unify_paging_params at
 0x7fdf17328048>
2019-08-12 12:36:18,326 - MainThread - botocore.loaders - DEBUG - Loading JSON
 file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
paginators-1.json
2019-08-12 12:36:18,326 - MainThread - awscli.customizations.paginate - DEBUG -
 Modifying paging parameters for operation: ListGroups
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_generate_skeleton at
 0x7fdf1737eae8>
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event
 before-building-argument-table-parser.iam.list-groups: calling handler
 <bound method OverrideRequiredArgsArgument.override_required_args of
 <awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event
 before-building-argument-table-parser.iam.list-groups: calling handler
 <bound method GenerateCliSkeletonArgument.override_required_args of
 <awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
 0x7fdf171c5978>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event operation-
args-parsed.iam.list-groups: calling handler functools.partial(<function
 check_should_enable_pagination at 0x7fdf17328158>, ['marker', 'max-items'], {'max-
items': <awscli.arguments.CLIArgument object at 0x7fdf171b09b0>}, OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
 <awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
```

```
 <awscli.customizations.paginate.PageArgument object at 0x7fdf171c58d0>), ('cli-
input-json', <awscli.customizations.cliinputjson.CliInputJSONArgument object at
 0x7fdf171b0a58>), ('starting-token', <awscli.customizations.paginate.PageArgument
 object at 0x7fdf171b0a20>), ('page-size', <awscli.customizations.paginate.PageArgument
 object at 0x7fdf171c5828>), ('generate-cli-skeleton',
 <awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
 0x7fdf171c5978>)]))
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.path-prefix: calling handler <awscli.paramfile.URIArgumentHandler
 object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.marker: calling handler <awscli.paramfile.URIArgumentHandler object
 at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.max-items: calling handler <awscli.paramfile.URIArgumentHandler
 object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
 Event load-cli-arg.iam.list-groups.cli-input-json: calling handler
 <awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
 Event load-cli-arg.iam.list-groups.starting-token: calling handler
 <awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.page-size: calling handler <awscli.paramfile.URIArgumentHandler
 object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event
 load-cli-arg.iam.list-groups.generate-cli-skeleton: calling handler
 <awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG
 - Event calling-command.iam.list-groups: calling handler
 <bound method CliInputJSONArgument.add_to_call_parameters of
 <awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG -
 Event calling-command.iam.list-groups: calling handler <bound
 method GenerateCliSkeletonArgument.generate_json_skeleton of
 <awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
 0x7fdf171c5978>>
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
 credentials via: assume-role
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
 credentials via: assume-role-with-web-identity
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
 credentials via: shared-credentials-file
```

```
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - INFO - Found credentials
 in shared credentials file: ~/.aws/credentials
2019-08-12 12:36:18,330 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/endpoints.json
2019-08-12 12:36:18,334 - MainThread - botocore.hooks - DEBUG - Event choose-service-
name: calling handler <function handle_service_name_alias at 0x7fdf1898eb70>
2019-08-12 12:36:18,337 - MainThread - botocore.hooks - DEBUG - Event creating-client-
class.iam: calling handler <function add_generate_presigned_url at 0x7fdf18a028c8>
2019-08-12 12:36:18,337 - MainThread - botocore.regions - DEBUG - Using partition
 endpoint for iam, us-west-2: aws-global
2019-08-12 12:36:18,337 - MainThread - botocore.args - DEBUG - The s3 config key is not
 a dictionary type, ignoring its value of: None
2019-08-12 12:36:18,340 - MainThread - botocore.endpoint - DEBUG - Setting iam timeout
 as (60, 60)
2019-08-12 12:36:18,341 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/_retry.json
2019-08-12 12:36:18,341 - MainThread - botocore.client - DEBUG - Registering retry
 handlers for service: iam
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
parameter-build.iam.ListGroups: calling handler <function generate_idempotent_uuid at
 0x7fdf189b10d0>
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
call.iam.ListGroups: calling handler <function inject_api_version_header_if_needed at
 0x7fdf189b2a60>
2019-08-12 12:36:18,343 - MainThread - botocore.endpoint - DEBUG - Making
 request for OperationModel(name=ListGroups) with params: {'url_path': '/',
 'query_string': '', 'method': 'POST', 'headers': {'Content-Type': 'application/x-
www-form-urlencoded; charset=utf-8', 'User-Agent': 'aws-cli/1.16.215 Python/3.7.3
 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205'}, 'body': {'Action':
 'ListGroups', 'Version': '2010-05-08'}, 'url': 'https://iam.amazonaws.com/',
 'context': {'client_region': 'aws-global', 'client_config': <botocore.config.Config
 object at 0x7fdf16e9a4a8>, 'has_streaming_input': False, 'auth_type': None}}
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event request-
created.iam.ListGroups: calling handler <bound method RequestSigner.handler of
 <botocore.signers.RequestSigner object at 0x7fdf16e9a470>>
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event choose-
signer.iam.ListGroups: calling handler <function set_operation_specific_signer at
 0x7fdf18996f28>
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - Calculating signature
 using v4 auth.
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - CanonicalRequest:
POST
/
```

```
content-type:application/x-www-form-urlencoded; charset=utf-8
host:iam.amazonaws.com
x-amz-date:20190812T193618Z

content-type;host;x-amz-date
5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEca69080e1e1f
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - StringToSign:
AWS4-HMAC-SHA256
20190812T193618Z
20190812/us-east-1/iam/aws4_request
ab7e367eEXAMPLE2769f178ea509978cf8bfa054874b3EXAMPLE8d043fab6cc9
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - Signature:
d85a0EXAMPLEb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
2019-08-12 12:36:18,344 - MainThread - botocore.endpoint - DEBUG - Sending
 http request: <AWSPreparedRequest stream_output=False, method=POST,
 url=https://iam.amazonaws.com/, headers={'Content-Type': b'application/
x-www-form-urlencoded; charset=utf-8', 'User-Agent': b'aws-cli/1.16.215
 Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205',
 'X-Amz-Date': b'20190812T193618Z', 'Authorization': b'AWS4-HMAC-SHA256
 Credential=AKIA01234567890EXAMPLE-east-1/iam/aws4_request, SignedHeaders=content-
type;host;x-amz-date, Signature=d85a07692aceb401EXAMPLEa1b18ad1ddf58a1a3ce7EXAMPLE',
 'Content-Length': '36'}>
2019-08-12 12:36:18,344 - MainThread - urllib3.util.retry - DEBUG - Converted retries
 value: False -> Retry(total=False, connect=None, read=None, redirect=0, status=None)
2019-08-12 12:36:18,344 - MainThread - urllib3.connectionpool - DEBUG - Starting new
 HTTPS connection (1): iam.amazonaws.com:443
2019-08-12 12:36:18,664 - MainThread - urllib3.connectionpool - DEBUG - https://
iam.amazonaws.com:443 "POST / HTTP/1.1" 200 570
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response headers:
 {'x-amzn-RequestId': '74c11606-bd38-11e9-9c82-559da0adb349', 'Content-Type': 'text/
xml', 'Content-Length': '570', 'Date': 'Mon, 12 Aug 2019 19:36:18 GMT'}
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response body:
b'<ListGroupsResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">\n
 <ListGroupsResult>\n    <IsTruncated>false</IsTruncated>\n    <Groups>\n
  <member>\n        <Path>/</Path>\n        <GroupName>MyTestGroup</GroupName>
\n        <Arn>arn:aws:iam::123456789012:group/MyTestGroup</Arn>\n
 <GroupId>AGPA1234567890EXAMPLE</GroupId>\n        <CreateDate>2019-08-12T19:34:04Z</
CreateDate>\n      </member>\n    </Groups>\n  </ListGroupsResult>\n
 <ResponseMetadata>\n    <RequestId>74c11606-bd38-11e9-9c82-559da0adb349</RequestId>\n
 </ResponseMetadata>\n</ListGroupsResponse>\n'
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event needs-
retry.iam.ListGroups: calling handler <botocore.retryhandler.RetryHandler object at
 0x7fdf16e9a780>
2019-08-12 12:36:18,665 - MainThread - botocore.retryhandler - DEBUG - No retry needed.
```

```
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event after-
call.iam.ListGroups: calling handler <function json_decode_policies at 0x7fdf189b1d90>
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "MyTestGroup",
            "GroupId": "AGPA123456789012EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
            "CreateDate": "2019-08-12T19:34:04Z"
        }
    ]
}
```

Retour en haut de page

Activer et consulter les journaux d'historique des AWS CLI commandes

Vous pouvez activer les journaux d'historique des AWS CLI commandes à l'aide du paramètre de `cli_history` fichier. Après avoir activé ce paramètre, AWS CLI enregistre l'historique des `aws` commandes.

Vous pouvez ainsi répertorier votre historique à l'aide de la `aws history list` commande et utiliser les command_ids obtenus dans la `aws history show` commande pour plus de détails. Pour plus d'informations, consultez le guide AWS CLI de référence.

Lorsque vous incluez l'`--debug`option, certains des détails incluent :

- Appels d'API effectués vers botocore
- Codes d’état
- Réponses HTTP
- En-têtes
- Codes de retour

Vous pouvez utiliser ces informations pour confirmer que les données des paramètres et les appels d'API se comportent comme prévu, puis déduire à quelle étape du processus votre commande échoue.

Retour en haut de page

Confirmez AWS CLI que votre

Diverses erreurs peuvent se produire si vos `credentials` fichiers `config` et ou votre utilisateur ou rôle IAM ne sont pas correctement configurés. Pour plus d'informations sur la résolution des erreurs liées aux `credentials` fichiers `config` et ou à votre utilisateur ou à vos rôles IAM, consultez the section called “Erreurs d'accès refusé” etthe section called “Informations d'identification non valides et erreurs clés”.

Retour en haut de page

Erreurs de commande non trouvées

Cette erreur signifie que le système d'exploitation ne trouve pas la AWS CLI commande. L'installation est peut-être incomplète ou nécessite une mise à jour.

Cause possible : vous essayez d'utiliser une AWS CLI fonctionnalité plus récente que la version installée, ou vous avez un formatage incorrect

Exemple de texte d'erreur :

```
$ aws s3 copy
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                                       | website
cp                                       | mv
....
```

Diverses erreurs peuvent se produire si votre commande est mal formatée ou si vous utilisez une version antérieure à la publication de la fonctionnalité. Pour plus d'informations sur la résolution des erreurs liées à ces deux problèmes, consultez the section called “Vérifiez le formatage AWS CLI de vos commandes” etthe section called “Vérifiez que vous utilisez une version récente du AWS CLI”.

Retour en haut de page

Cause possible : le terminal doit être redémarré après l'installation

Exemple de texte d'erreur :

```
$ aws --version
command not found: aws
```

Si la `aws` commande est introuvable après l'avoir installée ou mise à jour pour la première fois AWS CLI, vous devrez peut-être redémarrer votre terminal pour qu'il reconnaisse les `PATH` mises à jour.

Retour en haut de page

Cause possible : L'installation AWS CLI n'a pas été complète

Exemple de texte d'erreur :

```
$ aws --version
command not found: aws
```

Si la `aws` commande est introuvable après l'installation ou la mise à jour initiale AWS CLI, il se peut qu'elle n'ait pas été complètement installée. Essayez de le réinstaller en suivant les étapes correspondant à votre plateforme dans the section called “Installation/mise à jour”.

Retour en haut de page

Cause possible : Le AWS CLI n'a pas d'autorisations (Linux)

Si la `aws` commande est introuvable après l'avoir installée ou mise à jour pour la AWS CLI première fois sous Linux, il est possible qu'elle ne dispose pas des `execute` autorisations nécessaires pour le dossier dans lequel elle a été installée. Exécutez la commande suivante avec le `PATH` pour votre AWS CLI installation, afin de fournir `chmod` des autorisations à AWS CLI :

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

Retour en haut de page

Cause possible : le système d'exploitation n'**`PATH`**a pas été mis à jour lors de l'installation

Exemple de texte d'erreur :

```
$ aws --version
command not found: aws
```

Vous allez peut-être devoir ajouter le fichier exécutable `aws` à la variable d'environnement `PATH` de votre système d'exploitation. Pour ajouter le AWS CLI à votre`PATH`, suivez les instructions ci-dessous pour votre système d'exploitation.

Linux and macOS

1. Recherchez le script de profil de votre shell dans votre répertoire utilisateur. Si vous n'êtes pas certain du shell utilisé, exécutez `echo $SHELL`.

   ```
   $ ls -a ~
   .  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
   ```

   - Bash – `.bash_profile`, `.profile`, ou `.bash_login`
   - Zsh – `.zshrc`
   - Tcsh – `.tcshrc`, `.cshrc`, ou `.login`
2. Ajoutez une commande d'exportation à votre script de profil. La commande suivante ajoute votre bin local à la `PATH` variable actuelle.

   ```
   export PATH=/usr/local/bin:$PATH
   ```
3. Rechargez le profil mis à jour dans votre session en cours.

   ```
   $ source ~/.bash_profile
   ```

Windows

1. Dans une invite de commande Windows, utilisez la `where` commande avec le `/R` *`path`* paramètre pour trouver l'emplacement du `aws` fichier. Les résultats renvoient tous les dossiers contenant`aws`.

   ```
   C:\> where /R c:\ aws
   c:\Program Files\Amazon\AWSCLIV2\aws.exe
   ...
   ```

   Par défaut, la AWS CLI version 2 se trouve dans :

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. Appuyez sur la touche Windows et entrez **`environment variables`**.
3. Dans la liste des suggestions, choisissez Modifier les variables d'environnement pour votre compte.
4. Choisissez PATH, puis Modifier.
5. Ajoutez le chemin que vous avez trouvé lors de la première étape dans le champ Valeur variable, par exemple,***`C:\Program Files\Amazon\AWSCLIV2\aws.exe`***.
6. Sélectionnez OK deux fois pour appliquer les nouveaux paramètres.
7. Fermez toute invite de commande en cours d'exécution et rouvrez une invite de commande.

Retour en haut de page

La commande « **`aws --version`** » renvoie une version différente de celle que vous avez installée

Votre terminal renvoie peut-être une valeur différente de AWS CLI celle PATH à laquelle vous vous attendiez.

Cause possible : le terminal doit être redémarré après l'installation

Si la `aws` commande affiche une version incorrecte, vous devrez peut-être redémarrer votre terminal pour qu'il reconnaisse les PATH mises à jour. Tous les terminaux ouverts doivent être fermés, pas seulement votre terminal actif.

Retour en haut de page

Cause possible : le système doit être redémarré après l'installation

Si la `aws` commande affiche la mauvaise version et que le redémarrage du terminal n'a pas fonctionné, vous devrez peut-être redémarrer votre système pour qu'il reconnaisse vos PATH mises à jour.

Retour en haut de page

Cause possible : vous disposez de plusieurs versions du AWS CLI

Si vous avez mis à jour AWS CLI et utilisé une méthode d'installation différente de celle de votre installation préexistante, plusieurs versions peuvent être installées. Par exemple, si vous utilisez Linux ou macOS `pip` pour votre installation actuelle, mais que vous avez essayé de procéder à la mise à jour à l'aide du fichier d'`.pkg`installation, cela peut provoquer des conflits, en particulier si vous `PATH` pointez vers l'ancienne version.

Pour résoudre ce problème, désinstallez toutes les versions de AWS CLI et effectuez une nouvelle installation.

Après avoir désinstallé toutes les versions, suivez les instructions adaptées à votre système d'exploitation pour installer la version souhaitée de la AWS CLI version 1 ou de la AWS CLI version 2.

> **Note**
>
> Si cela se produit après avoir installé la AWS CLI version 2 avec une installation préexistante de la AWS CLI version 1, suivez les instructions de migration figurant dans les 1. the section called “Instructions de migration”

Retour en haut de page

La commande « **`aws --version`** » renvoie une version après la désinstallation du AWS CLI

Cela se produit souvent lorsqu'il y en a encore un AWS CLI installé quelque part sur votre système.

Cause possible : le terminal doit être redémarré après la désinstallation

Si la `aws --version` commande fonctionne toujours, vous devrez peut-être redémarrer votre terminal pour qu'il reconnaisse les mises à jour du terminal.

Retour en haut de page

Cause possible : vous disposez de plusieurs versions de AWS CLI votre système ou vous n'avez pas utilisé la même méthode de désinstallation que celle que vous avez utilisée pour installer le AWS CLI

La désinstallation AWS CLI peut ne pas être correcte si vous l'avez désinstallée AWS CLI en utilisant une méthode différente de celle que vous avez utilisée pour l'installer, ou si vous avez installé plusieurs versions. Par exemple, si vous l'avez utilisé `pip` pour votre installation actuelle, vous devez l'utiliser `pip` pour le désinstaller. Pour résoudre ce problème, désinstallez-le AWS CLI en utilisant la même méthode que celle que vous avez utilisée pour l'installer.

1. Suivez les instructions correspondant à votre système d'exploitation et à votre méthode d'installation d'origine pour désinstaller les AWS CLI versions 1 et AWS CLI 2.
2. Fermez tous les terminaux que vous avez ouverts.
3. Ouvrez votre terminal préféré, entrez la commande suivante et confirmez qu'aucune version n'est renvoyée.

```
$ aws --version
command not found: aws
```

Si une version est toujours répertoriée dans la sortie, AWS CLI elle a probablement été installée à l'aide d'une autre méthode ou il existe plusieurs versions. Si vous ne savez pas quelle méthode vous avez installée AWS CLI, suivez les instructions pour chaque méthode de désinstallation pour les AWS CLI versions 1 et AWS CLI 2 appropriées à votre système d'exploitation jusqu'à ce qu'aucune sortie de version ne soit reçue.

> **Note**
>
> Si vous avez utilisé un gestionnaire de packages pour installer le AWS CLI (`pip`apt,`brew`,, etc.), vous devez utiliser le même gestionnaire de packages pour le désinstaller. Assurez-vous de suivre les instructions fournies par le gestionnaire de packages pour savoir comment désinstaller toutes les versions d'un package.

Retour en haut de page

Le a AWS CLI traité une commande avec un nom de paramètre incomplet

Cause possible : vous avez utilisé une abréviation reconnue du AWS CLI paramètre

Étant donné que le AWS CLI est construit en utilisant Python, il AWS CLI utilise la `argparse` bibliothèque Python, y compris l'`allow_abbrev`argument. Les abréviations des paramètres sont reconnues par le AWS CLI et traitées.

L'exemple de commande suivant modifie le nom de la CloudFormation pile. Le paramètre `--change-set-n` est reconnu comme une abréviation de`--change-set-name`, et AWS CLI traite la commande.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

Lorsque votre abréviation peut être composée de plusieurs commandes, le paramètre n'est pas reconnu comme une abréviation.

L'exemple de commande suivant modifie le nom de la CloudFormation pile. Le paramètre n'**`--change-set-`**est pas reconnu comme une abréviation, car il existe plusieurs paramètres dont il peut être une abréviation, tels que `--change-set-name` et`--change-set-type`. Par conséquent, la commande AWS CLI n'est pas traitée.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

> **Warning**
>
> N'utilisez pas délibérément des abréviations de paramètres. Ils ne sont pas fiables et ne sont pas rétrocompatibles. Si de nouveaux paramètres sont ajoutés à une commande qui confond vos abréviations, vos commandes seront annulées.
> En outre, si le paramètre est un argument à valeur unique, il peut entraîner un comportement inattendu de vos commandes. Si plusieurs instances d'un argument à valeur unique sont transmises, seule la dernière instance sera exécutée. Dans l'exemple suivant, le paramètre `--filters` est un argument à valeur unique. Les paramètres `--filters` et `--filter` sont spécifiés. Le `--filter` paramètre est l'abréviation de`--filters`. Cela entraîne l'application de `--filters` deux instances et seul le dernier `--filter` argument s'applique.

```
$ aws ec2 describe-vpc-peering-connections \
    --filters Name=tag:TagName,Values=VpcPeeringConnection \
    --filter Name=status-code,Values=active
```

Vérifiez que vous utilisez des paramètres valides avant d'exécuter une commande afin d'éviter tout comportement inattendu.

Retour en haut de page

Erreurs d'accès refusé

Cause possible : le fichier AWS CLI programme n'a pas l'autorisation « Exécuter »

Sous Linux ou macOS, assurez-vous que le `aws` programme dispose des autorisations d'exécution pour l'utilisateur appelant. En règle générale, les autorisations sont définies sur `755`.

Pour ajouter une autorisation d'exécution à votre utilisateur, exécutez la commande suivante en la *`~/.local/bin/aws`* remplaçant par le chemin d'accès au programme sur votre ordinateur.

```
$ chmod +x ~/.local/bin/aws
```

Retour en haut de page

Cause possible : votre identité IAM n'est pas autorisée à effectuer l'opération

Exemple de texte d'erreur :

```
$ aws s3 ls
An error occurred (AcessDenied) when calling the ListBuckets operation: Access
 denied.
```

Lorsque vous exécutez une AWS CLI commande, les AWS opérations sont effectuées en votre nom, à l'aide des informations d'identification qui vous associent à un compte ou à un rôle IAM. Les politiques associées doivent vous autoriser à appeler les actions d'API correspondant aux commandes que vous exécutez avec le AWS CLI.

La plupart des commandes appellent une seule action avec un nom correspondant au nom de la commande. Cependant, des commandes personnalisées telles que `aws s3 sync` l'appel multiple APIs. Vous pouvez voir quelle APIs commande appelle en utilisant l'`--debug`option.

Si vous êtes certain que l'utilisateur ou le rôle dispose des autorisations appropriées attribuées par la politique, assurez-vous que votre AWS CLI commande utilise les informations d'identification que vous attendez. Consultez la section suivante sur les informations d'identification pour vérifier que les AWS CLI informations d'identification utilisées sont celles que vous attendez.

Pour plus d'informations sur l'attribution d'autorisations IAM, voir Présentation de la gestion des accès : autorisations et politiques dans le guide de l'utilisateur IAM.

Retour en haut de page

Informations d'identification non valides et erreurs clés

Exemple de texte d'erreur :

```
$ aws s3 ls
An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS
 Access Key Id
you provided does not exist in our records.
```

```
$ aws s3 ls
An error occurred (InvalidClientTokenId) when calling the ListBuckets operation: The
 security token
included in the request is invalid.
```

Cause possible : les informations d'identification AWS CLI sont en train de lire des informations d'identification incorrectes ou proviennent d'un emplacement inattendu

AWS CLI Il est possible qu'ils lisent les informations d'identification à un endroit différent de celui prévu ou que les informations relatives à votre paire de clés soient incorrectes. Vous pouvez exécuter `aws configure list` pour confirmer quelles informations d'identification sont utilisées.

L'exemple suivant montre comment vérifier les informations d'identification utilisées pour le profil par défaut.

```
$ aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key     ****************XYVA shared-credentials-file
secret_key     ****************ZAGY shared-credentials-file
    region                us-west-2      config-file    ~/.aws/config
```

L'exemple suivant montre comment vérifier les informations d'identification utilisées pour un profil nommé.

```
$ aws configure list --profile saanvi
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                    saanvi           manual    --profile
access_key         *************** shared-credentials-file
secret_key         *************** shared-credentials-file
    region                us-west-2      config-file    ~/.aws/config
```

Pour confirmer les détails de votre paire de clés, consultez vos `credentials` fichiers `config` et. Pour plus d'informations sur `credentials` les fichiers `config` et, consultezthe section called “Paramètres de configuration”. Pour plus d'informations sur les informations d'identification et l'authentification, y compris la priorité des informations d'identification, consultezAuthentification et identifiants d'accès.

Retour en haut de page

Cause possible : l'horloge de votre ordinateur n'est pas synchronisée

Si vous utilisez des informations d'identification valides, votre horloge est peut-être désynchronisée. Sur Linux ou macOS, exécutez `date` pour vérifier l'heure.

```
$ date
```

Si l'horloge de votre système n'est pas exacte en quelques minutes, utilisez `ntpd` pour la synchroniser.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Sous Windows, utilisez les options de date et d'heure du Panneau de configuration pour configurer votre horloge système.

Retour en haut de page

La signature ne correspond pas aux erreurs

Exemple de texte d'erreur :

```
$ aws s3 ls
An error occurred (SignatureDoesNotMatch) when calling the ListBuckets operation: The
 request signature we
calculated does not match the signature you provided. Check your key and signing
 method.
```

Lorsque le AWS CLI exécute une commande, il envoie une demande cryptée aux AWS serveurs pour effectuer les opérations de AWS service appropriées. Vos informations d'identification (clé d'accès et clé secrète) sont impliquées dans le chiffrement et permettent AWS d'authentifier la personne qui fait la demande. Plusieurs éléments peuvent interférer avec le bon fonctionnement de ce processus, comme suit :

Cause possible : votre horloge n'est pas synchronisée avec les AWS serveurs

Pour vous protéger contre les attaques en replay, l'heure actuelle peut être utilisée pendant le encryption/decryption processus. Si la différence d'heure entre le client et le serveur est supérieure à la valeur autorisée, le processus peut échouer et la demande est rejetée. Cela peut également se produire lorsque vous exécutez une commande dans une machine virtuelle dont l'horloge est désynchronisée par rapport à l'horloge de la machine hôte. L'une des causes possibles est le fait que la machine virtuelle hiberne et qu'il faut un certain temps après le réveil pour synchroniser l'horloge avec la machine hôte.

Sur Linux ou macOS, exécutez `date` pour vérifier l'heure.

```
$ date
```

Si l'horloge de votre système n'est pas exacte en quelques minutes, utilisez `ntpd` pour la synchroniser.

```
$ sudo service ntpd stop
```

```
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Sous Windows, utilisez les options de date et d'heure du Panneau de configuration pour configurer votre horloge système.

Retour en haut de page

Cause possible : votre système d'exploitation ne gère pas correctement AWS les touches contenant certains caractères spéciaux

Si vos AWS touches contiennent certains caractères spéciaux, tels que,`-`, `+` `/`%, ou si certaines variantes du système d'exploitation traitent la chaîne de manière incorrecte, ce qui entraîne une interprétation incorrecte de la chaîne de touches.

Si vous traitez vos clés à l'aide d'autres outils ou scripts, tels que des outils qui créent le fichier d'informations d'identification sur une nouvelle instance dans le cadre de sa création, ces outils et scripts peuvent avoir leur propre gestion des caractères spéciaux, ce qui les transforme en quelque chose qui AWS ne les reconnaît plus.

Nous vous suggérons de régénérer la clé secrète pour en obtenir une qui n'inclut pas le caractère spécial à l'origine du problème.

Retour en haut de page

Erreurs de certificat SSL

Cause possible : le certificat de votre proxy AWS CLI n'est pas fiable

Exemple de texte d'erreur :

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Lorsque vous utilisez une AWS CLI commande, vous recevez un message `[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed` d'erreur. Cela est dû au fait que vous AWS CLI ne faites pas confiance au certificat de votre proxy en raison de facteurs tels que le certificat de votre proxy étant auto-signé, votre entreprise étant définie comme autorité de certification (CA). Cela empêche AWS CLI de trouver le certificat racine de l'autorité de certification de votre entreprise dans le registre local de l'autorité de certification.

Pour résoudre ce problème, indiquez AWS CLI où trouver le fichier de votre entreprise à l'aide du paramètre du `.pem` fichier de `ca_bundle` configuration, de l'option de ligne de `--ca-bundle` commande ou de la variable d'`AWS_CA_BUNDLE`environnement.

Retour en haut de page

Cause possible : votre configuration ne pointe pas vers le bon emplacement du certificat racine de l'autorité de certification

Exemple de texte d'erreur :

```
$ aws s3 ls
SSL validation failed for regionname [Errno 2] No such file or directory
```

Cela est dû au fait que l'emplacement du fichier de votre bundle d'autorité de certification (CA) n'est pas correctement configuré dans le AWS CLI. Pour résoudre ce problème, vérifiez où se trouve `.pem` le fichier de votre entreprise et mettez à jour la AWS CLI configuration à l'aide du paramètre du fichier de `ca_bundle` configuration, de l'option de ligne de `--ca-bundle` commande ou de la variable d'`AWS_CA_BUNDLE`environnement.

Retour en haut de page

Cause possible : votre configuration n'utilise pas le bon Région AWS

Exemple de texte d'erreur :

```
$ aws s3 ls
[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed
```

Des erreurs ou des résultats inattendus peuvent se produire si un Service AWS n'est pas disponible pour ce que vous avez spécifié Région AWS ou si vos ressources se trouvent dans un autre endroit Région AWS. Pour obtenir les étapes de dépannage, consultez the section called “Vérifiez que Région AWS votre AWS CLI commande utilise”.

Retour en haut de page

Cause possible : votre version TLS doit être mise à jour

Exemple de texte d'erreur :

```
$ aws s3 ls
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

Service AWS Il utilise une version de TLS incompatible avec la version TLS de votre appareil. Pour résoudre ce problème, effectuez une mise à jour vers une version TLS prise en charge. Pour de plus amples informations, veuillez consulter the section called “Application d'une version minimale de TLS”.

Retour en haut de page

Erreurs JSON non valides

Exemple de texte d'erreur :

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,WriteCapacityUnits":10}' \
    --table-name MyDDBTable
Error parsing parameter '--provisioned-throughput': Invalid JSON: Expecting property
 name enclosed in
double quotes: line 1 column 25 (char 24)
JSON received: {"ReadCapacityUnits":15,WriteCapacityUnits":10}
```

Lorsque vous utilisez une AWS CLI commande, vous recevez un message d'erreur `Invalid JSON` « ». Il s'agit généralement d'une erreur qui se produit lorsque vous entrez une commande au format JSON attendu et que vous AWS CLI ne pouvez pas lire correctement votre JSON.

Cause possible : Vous n'avez pas saisi un code JSON valide AWS CLI pour l'utiliser

Vérifiez que vous avez saisi un code JSON valide pour votre commande. Nous vous suggérons d'utiliser un validateur JSON pour les fichiers JSON que vous rencontrez des problèmes de formatage.

Pour une utilisation plus avancée du JSON dans la ligne de commande, pensez à utiliser un processeur JSON en ligne de commande`jq`, par exemple pour créer des chaînes JSON. Pour plus d'informations sur`jq`, consultez le référentiel jq sur GitHub.

Retour en haut de page

Cause possible : les règles de citation de votre terminal empêchent l'envoi d'un JSON valide au AWS CLI

Avant de AWS CLI recevoir quoi que ce soit d'une commande, votre terminal traite la commande en utilisant ses propres règles de citation et d'échappement. En raison des règles de formatage d'un terminal, une partie de votre contenu JSON peut être supprimée avant que la commande ne

soit transmise au AWS CLI. Lorsque vous formulez des commandes, veillez à utiliser les règles de citation de votre terminal.

Pour résoudre le problème, utilisez la `echo` commande pour voir comment le shell gère vos paramètres :

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}
ReadCapacityUnits:15 WriteCapacityUnits:10
```

```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

Modifiez votre commande jusqu'à ce que votre code JSON valide soit renvoyé.

Pour un dépannage plus approfondi, utilisez le `--debug` paramètre pour afficher les journaux de débogage, car ils afficheront exactement ce qui a été transmis au AWS CLI :

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,WriteCapacityUnits":10}' \
    --table-name MyDDBTable \
    --debug
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.18.147
Python/2.7.18 Linux/5.4.196-119.356.amzn2int.x86_64 botocore/1.18.6
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - Arguments entered
 to CLI:
['dynamodb', 'update-table', '--provisioned-throughput',
 '{"ReadCapacityUnits":15,WriteCapacityUnits":10}',
 '--table-name', 'MyDDBTable', '--debug']
```

Utilisez les règles de citation de votre terminal pour résoudre les problèmes rencontrés par votre entrée JSON lorsqu'elle est envoyée au AWS CLI. Pour plus d'informations sur les règles de citation, voirthe section called “Citations avec chaînes”.

> **Note**
>
> Si vous rencontrez des problèmes pour obtenir un code JSON valide dans le AWS CLI, nous vous recommandons de contourner les règles de citation d'un terminal pour la saisie de données JSON en utilisant des blobs pour transmettre vos données JSON directement au. AWS CLI Pour plus d'informations sur les blobs, consultezBLOB.

Retour en haut de page

Ressources supplémentaires

Pour obtenir de l'aide supplémentaire AWS CLI concernant vos problèmes, consultez la AWS CLI communauté GitHubou la AWS re:Post communauté.

Retour en haut de page

Exemples pour le AWS CLI

Ce chapitre fournit une collection d'exemples qui vous montrent comment utiliser le AWS Command Line Interface (AWS CLI) avec Services AWS.

Ce guide AWS CLI contient les types d'exemples suivants :

- Exemples de commandes guidées - Exemples de commandes guidées pour le guide de AWS CLI l'utilisateur expliquant comment utiliser le AWS CLI avec certaines Services AWS. Il s'agit souvent d'exemples plus détaillés que ceux du version 2 du guide de référence.
- AWS CLI exemples de commandes - Exemples de commandes open source également disponibles dans le version 2 du guide de référence. Les exemples de commandes sont hébergés dans le AWS CLIréférentiel sur GitHub.
- AWS CLI utilisation d'exemples de code de script Bash - Exemples de scripts bash open source. Les exemples de scripts Bash sont hébergés dans le référentiel d'exemples de AWS code sur. GitHub

> **Exemple de feedback**
>
> Vous n’avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de commande en utilisant le lien Fournir des commentaires au bas de cette page ou sur la page de commande correspondante dans le version 2 du guide de référence.
> Vous souhaitez apporter votre contribution ? Fournissez des exemples de AWS CLI commandes dans le référentiel d'exemples de AWS code sur GitHub. Pour plus d'informations sur la contribution, consultez les exemples de AWS CLI code sur les étapes rapides de contribution sur GitHubles pages.

Exemples de commandes guidées pour AWS CLI

The AWS Command Line Interface (AWS CLI) est un outil open source qui vous permet d'interagir avec un large éventail de commandes d' Services AWS utilisation dans votre interface de ligne de commande. Cette section fournit des exemples guidés qui montrent comment tirer parti du AWS CLI pour accéder à certains des Services AWS. Cela inclut certaines AWS CLI commandes personnalisées, telles que les `aws s3` commandes de haut niveau. Ces exemples de commandes

illustrent les actions courantes utilisées pour certaines Services AWS et fournissent des ressources supplémentaires pour plus d'informations.

Que vous soyez un AWS utilisateur expérimenté ou un débutant AWS CLI, ces exemples guidés constituent une ressource pour rationaliser vos AWS opérations.

Pour une référence complète de toutes les commandes disponibles pour chacune Service AWS, consultez le version 2 du guide de référence. En outre, vous pouvez utiliser l'aide de ligne de commande intégrée pour explorer la gamme de commandes Services AWS, d'options et de fonctionnalités du AWS CLI.

Pour d'autres exemples de commandes qui ne sont pas disponibles dans cette section, consultez la AWS CLI exemples de commandes section. Ce sont des exemples de commandes open source qui sont également disponibles dans le version 2 du guide de référence. Les exemples de commandes sont hébergés dans le AWS CLIréférentiel sur GitHub.

Pour des exemples de scripts bash open source, voir. the section called “Exemples de scripts Bash” Les exemples de scripts Bash sont hébergés dans le référentiel d'exemples de AWS code sur. GitHub

Services

- Utilisation d'Amazon DynamoDB dans AWS CLI
- Utilisation d'Amazon EC2 dans le AWS CLI
- Utilisation d'Amazon S3 Glacier dans AWS CLI
- Utilisation d'IAM dans AWS CLI
- Utilisation d'Amazon S3 dans AWS CLI
- Accès à Amazon SNS dans le AWS CLI

Utilisation d'Amazon DynamoDB dans AWS CLI

Le AWS Command Line Interface (AWS CLI) prend en charge tous les services de AWS base de données, y compris Amazon DynamoDB. Vous pouvez l'utiliser AWS CLI pour des opérations impromptues, telles que la création d'une table. Vous pouvez également l'utiliser pour intégrer des opérations DynamoDB dans des scripts utilitaires.

Pour plus d'informations sur l'utilisation de AWS CLI avec DynamoDB, *dynamodb* consultez le manuel de référence des commandes.AWS CLI

Pour répertorier les AWS CLI commandes de DynamoDB, utilisez la commande suivante.

```
$ aws dynamodb help
```

Rubriques

- Prérequis
- Création et utilisation de tables DynamoDB
- Utilisation de DynamoDB Local
- Ressources

Prérequis

Pour exécuter les dynamodb commandes, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.

Création et utilisation de tables DynamoDB

Le format de ligne de commande consiste en un nom de commande DynamoDB, suivi des paramètres de cette commande. Il AWS CLI prend en charge la syntaxe abrégée de la CLI pour les valeurs des paramètres et le JSON complet.

L'exemple suivant crée une table nommée `MusicCollection`.

```
$ aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions AttributeName=Artist,AttributeType=S
 AttributeName=SongTitle,AttributeType=S \
    --key-schema AttributeName=Artist,KeyType=HASH
 AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1
```

Vous pouvez ajouter de nouvelles lignes à la table avec des commandes similaires à celles indiquées dans l'exemple suivant. Ces exemples utilisent une combinaison de la syntaxe raccourcie et de JSON.

```
$ aws dynamodb put-item \
```

```
    --table-name MusicCollection \
    --item '{
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"} ,
        "AlbumTitle": {"S": "Somewhat Famous"}
      }' \
    --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
        "CapacityUnits": 1.0,
        "TableName": "MusicCollection"
    }
}
```

```
$ aws dynamodb put-item \
    --table-name MusicCollection \
    --item '{
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"} ,
        "AlbumTitle": {"S": "Songs About Life"}
      }' \
    --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
        "CapacityUnits": 1.0,
        "TableName": "MusicCollection"
    }
}
```

Il peut être difficile de composer un JSON valide sur une seule ligne de commande. Pour faciliter les choses, ils AWS CLI peuvent lire les fichiers JSON. Par exemple, imaginons l'extrait JSON suivant qui est stocké dans un fichier nommé `expression-attributes.json`.

```
{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Call Me Today"}
}
```

Vous pouvez utiliser ce fichier pour émettre une demande `query` à l'aide de l' AWS CLI. Dans l'exemple suivant, le contenu du fichier `expression-attributes.json` est utilisé pour le paramètre `--expression-attribute-values`.

```
$ aws dynamodb query --table-name MusicCollection \
    --key-condition-expression "Artist = :v1 AND SongTitle = :v2" \
    --expression-attribute-values file://expression-attributes.json
{
    "Count": 1,
    "Items": [
        {
            "AlbumTitle": {
                "S": "Somewhat Famous"
            },
            "SongTitle": {
                "S": "Call Me Today"
            },
            "Artist": {
                "S": "No One You Know"
            }
        }
    ],
    "ScannedCount": 1,
    "ConsumedCapacity": null
}
```

Utilisation de DynamoDB Local

Outre DynamoDB, vous pouvez utiliser le AWS CLI avec DynamoDB Local. DynamoDB Local est une petite base de données et un serveur côté client qui imite le service DynamoDB. DynamoDB Local vous permet d'écrire des applications qui utilisent l'API DynamoDB, sans manipuler les tables ou les données du service Web DynamoDB. Au lieu de cela, toutes les actions d'API sont redirigées vers une base de données locale. Cela vous permet d'économiser sur le débit alloué, le stockage de données et les frais de transfert de données.

Pour plus d'informations sur DynamoDB Local et sur son utilisation avec AWS CLI le, consultez les sections suivantes du manuel Amazon DynamoDB Developer Guide :

- DynamoDB Local
- Utilisation du AWS CLI avec DynamoDB Local

Ressources

AWS CLI référence :

- `aws dynamodb`
- `aws dynamodb create-table`
- `aws dynamodb put-item`
- `aws dynamodb query`

Référence du service :

- DynamoDB Local dans le guide du développeur Amazon DynamoDB
- Utilisation de AWS CLI DynamoDB Local dans le manuel du développeur Amazon DynamoDB

Utilisation d'Amazon EC2 dans le AWS CLI

Présentation d'Amazon Elastic Compute Cloud
Présentation d'Amazon EC2 - Elastic Cloud Server et hébergement avec AWS

Amazon Elastic Compute Cloud (Amazon EC2) fournit des environnements informatiques virtuels hautement évolutifs et flexibles. Amazon vous EC2 permet de provisionner et de gérer des serveurs virtuels, appelés EC2 instances Amazon, afin de répondre à un large éventail de besoins informatiques.

Les EC2 instances Amazon sont des machines virtuelles qui peuvent être personnalisées avec différentes configurations de processeur, de mémoire, de stockage et de capacités réseau. Vous pouvez choisir parmi une sélection variée de types d'instances, allant d'options légères et économiques à des instances puissantes et performantes, en fonction des exigences de votre application. Cette flexibilité vous permet de répondre à vos besoins informatiques afin d'optimiser les performances et la rentabilité.

En outre, Amazon EC2 propose une suite de fonctionnalités qui vous permettent de gérer efficacement vos ressources informatiques. Il s'agit notamment de la possibilité de lancer rapidement de nouvelles instances, de créer des images de machine personnalisées (AMIs) pour un déploiement rapide et d'augmenter ou de réduire la capacité de calcul selon les besoins.

Vous pouvez accéder aux fonctionnalités d'Amazon EC2 à l'aide du AWS Command Line Interface (AWS CLI). Pour répertorier les AWS CLI commandes pour Amazon EC2, utilisez la commande suivante.

```
aws ec2 help
```

Avant d'exécuter des commandes, définissez vos informations d'identification par défaut. Pour de plus amples informations, veuillez consulter Configuration des paramètres pour AWS CLI.

Cette rubrique présente de courts exemples de AWS CLI commandes qui exécutent des tâches courantes pour Amazon EC2.

Pour des exemples détaillés de AWS CLI commandes, consultez le référentiel d'exemples de AWS CLI code sur GitHub.

Rubriques

- Création, affichage et suppression de paires de EC2 clés Amazon dans AWS CLI
- Création, configuration et suppression de groupes EC2 de sécurité Amazon dans AWS CLI
- Lancement, mise en vente et suppression d' EC2instances Amazon dans AWS CLI
- Modifiez le type d' EC2 instance Amazon à l'aide d'un script bash dans le AWS CLI

Création, affichage et suppression de paires de EC2 clés Amazon dans AWS CLI

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer, afficher et supprimer vos paires de clés pour Amazon Elastic Compute Cloud (Amazon EC2). Vous utilisez des paires de clés pour vous connecter à une EC2 instance Amazon. Vous pouvez fournir la paire de clés à Amazon EC2 lorsque vous créez l'instance, puis utiliser cette paire de clés pour vous authentifier lorsque vous vous connectez à l'instance.

> **Note**
>
> Pour des exemples de commandes supplémentaires, consultez le de référence.

Rubriques

- Prérequis
- Création d’une paire de clés
- Afficher votre paire de clés
- Supprimer votre paire de clés

- Références

Prérequis

Pour exécuter les `ec2` commandes, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Définissez vos autorisations IAM pour autoriser l' EC2 accès à Amazon. Pour plus d'informations sur les autorisations IAM pour Amazon EC2, consultez les politiques IAM pour Amazon EC2 dans le guide de EC2 l'utilisateur Amazon.

Création d'une paire de clés

Pour créer une paire de clés, utilisez la `aws ec2 create-key-pair` commande avec l'`--query`option et l'`--output text`option pour transférer votre clé privée directement dans un fichier.

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
 > MyKeyPair.pem
```

En effet PowerShell, la `> file` redirection utilise par défaut le codage UTF-8, qui ne peut pas être utilisé avec certains clients SSH. Par conséquent, vous devez convertir la sortie en la transférant vers la commande `out-file` et explicitement définir l'encodage pour `ascii`.

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
 | out-file -encoding ascii -filepath MyKeyPair.pem
```

Le fichier `MyKeyPair.pem` qui en résulte ressemble à ce qui suit.

```
-----BEGIN RSA PRIVATE KEY-----
EXAMPLEKEYKCAQEAy7WZhaDsrA1W3mRlQtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7iO5dSrvC7dQkW2duV5QuUdE0QW
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449ilx9X1F
G50TCFeOzfl8dqqCP6GzbPaIjiU19xX/azOR9V+tpUOzEL+wmXnZt3/nHPQ5xvD2OJH67km6SuPW
oPzev/D8V+x4+bHthfSjR9Y7DvQFjfBVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrqu
/uler7vgIn5m7lN5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/ufGxbL1
mb5qwMGUnEpJaZD6QSSs3kICLwWUYUiGfc0uiSbmJoap/GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2
bahyWyJNfjLe4M86yd2YK3V2CmK+X/BOsShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/tJWSD9
81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA3OzdXzMQexXVJ1TLZVEH0E7bhlY9d8O1ozR
```

```
oQs/FiZNAx2iijCWyv0lpjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfql+lIp1
YkriL0DbLXlvRAH+yHPRit2hHOjtUNZh4Axv+cpg09qbUI3+43eEy24B7G/Uh+GTfbjsXsOxQx/x
p9otyVwc7hsQ5TA5PZb+mvkJ5OBEKzet9XcKwONBYELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
ayav0elc5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URrpzWVOEihvm+xTtmaZlSp//lkq75XDwnU
WA8gkn6O3QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
gYBjbO+OZk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY7Ozd5wQewBQ4AdSlWSX4nGDtsiFxWiI5sKuAAeOCbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vvwKBgF+09VI/1wJBirsDGz9whVWfFPrTkJNvJZzYt69qezxlsjgFKshy
WBhd4xHZtmCqpBPlAymEjr/TOlbxyARmXMnIOWIAnNXMGB4KGSyl1mzSVAoQ+fqR+cJ3d0dyPl1j
jjb0Ed/NY8frlNDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0iOegLDa
NWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvtW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asrnF3qS
VRkAKKKYeGjkpUfVTrW0YFjXkfcrR/V+QFL5OndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----
```

Votre clé privée n'est pas stockée AWS et ne peut être récupérée que lorsqu'elle est créée. Vous ne pouvez pas la récupérer plus tard. Au lieu de cela, si vous perdez votre clé privée, vous devez créer une nouvelle paire de clés.

Si vous vous connectez à votre instance depuis un ordinateur Linux, nous vous recommandons d'utiliser la commande suivante pour définir les autorisations de votre fichier de clé privée afin d'être la seule personne autorisée à le lire.

```
$ chmod 400 MyKeyPair.pem
```

Afficher votre paire de clés

Une « empreinte digitale » est générée à partir de votre paire de clés, et vous pouvez l'utiliser pour vérifier que la clé privée que vous avez sur votre machine locale correspond à la clé publique qui y est stockée AWS.

L'empreinte digitale est un SHA1 hachage extrait d'une copie codée DER de la clé privée. Cette valeur est capturée lors de la création de la paire de clés et est stockée AWS avec la clé publique. Vous pouvez consulter l'empreinte digitale dans la EC2 console Amazon ou en exécutant la AWS CLI commande`aws ec2 describe-key-pairs`.

L'exemple suivant affiche l'empreinte digitale pour `MyKeyPair`.

```
$ aws ec2 describe-key-pairs --key-name MyKeyPair
{
    "KeyPairs": [
        {
```

```
            "KeyName": "MyKeyPair",
            "KeyFingerprint":
 "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f"
        }
    ]
}
```

Pour plus d'informations sur les clés et les empreintes digitales, consultez Amazon EC2 Key Pairs dans le guide de l' EC2 utilisateur Amazon.

Supprimer votre paire de clés

Pour supprimer une paire de clés, exécutez la `aws ec2 delete-key-pair` commande en la *`MyKeyPair`* remplaçant par le nom de la paire à supprimer.

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

Références

AWS CLI référence :

- `aws ec2`
- `aws ec2 create-key-pair`
- `aws ec2 delete-key-pair`
- `aws ec2 describe-key-pairs`

Autre référence :

- Documentation Amazon Elastic Compute Cloud
- Pour consulter le AWS SDK et les exemples de AWS CLI code et y contribuer, consultez le référentiel d'exemples de AWS code sur GitHub.

Création, configuration et suppression de groupes EC2 de sécurité Amazon dans AWS CLI

Vous pouvez créer un groupe de sécurité pour vos instances Amazon Elastic Compute Cloud (Amazon EC2) qui fonctionne essentiellement comme un pare-feu, avec des règles qui déterminent le trafic réseau qui peut entrer et sortir.

Utilisez le AWS Command Line Interface (AWS CLI) pour créer un groupe de sécurité, ajouter des règles aux groupes de sécurité existants et supprimer des groupes de sécurité.

> **Note**
>
> Pour des exemples de commandes supplémentaires, consultez le de référence.

Rubriques

- Prérequis
- Création d'un groupe de sécurité
- Ajouter des règles à votre groupe de sécurité
- Supprimer votre groupe de sécurité
- Références

Prérequis

Pour exécuter les `ec2` commandes, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Définissez vos autorisations IAM pour autoriser l' EC2 accès à Amazon. Pour plus d'informations sur les autorisations IAM pour Amazon EC2, consultez les politiques IAM pour Amazon EC2 dans le guide de EC2 l'utilisateur Amazon.

Création d'un groupe de sécurité

Vous pouvez créer des groupes de sécurité associés à des clouds privés virtuels (VPCs).

L'`aws ec2 create-security-group`exemple suivant montre comment créer un groupe de sécurité pour un VPC spécifique.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --
vpc-id vpc-1a2b3c4d
{
    "GroupId": "sg-903004f8"
}
```

Pour afficher les informations initiales d'un groupe de sécurité, exécutez la `aws ec2 describe-security-groups` commande. Vous ne pouvez référencer un groupe de sécurité EC2 -VPC que par son nom`vpc-id`, et non par son nom.

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": []
                }
            ],
            "Description": "My security group"
            "IpPermissions": [],
            "GroupName": "my-sg",
            "VpcId": "vpc-1a2b3c4d",
            "OwnerId": "123456789012",
            "GroupId": "sg-903004f8"
        }
    ]
}
```

Ajouter des règles à votre groupe de sécurité

Lorsque vous exécutez une EC2 instance Amazon, vous devez activer les règles dans le groupe de sécurité afin d'autoriser le trafic réseau entrant pour votre moyen de connexion à l'image.

Par exemple, si vous lancez une instance Windows, vous ajoutez généralement une règle pour autoriser le trafic entrant sur le port TCP 3389 afin de prendre en charge le protocole RDP (Remote Desktop Protocol). Si vous lancez une instance Linux, vous ajoutez généralement une règle pour autoriser le trafic entrant sur le port TCP 22 afin de prendre en charge les connexions SSH.

Utilisez la commande `aws ec2 authorize-security-group-ingress` pour ajouter une règle à votre groupe de sécurité. L'un des paramètres obligatoires de cette commande est l'adresse

IP publique de votre ordinateur ou le réseau (sous la forme d'une plage d'adresses) auquel votre ordinateur est attaché, en notation CIDR.

> **Note**
>
> Nous fournissons le service suivant, https://checkip.global.api.aws/, pour vous permettre de déterminer votre adresse IP publique. Pour trouver d'autres services qui peuvent vous aider à identifier votre adresse IP, recherchez « quelle est mon adresse IP » sur votre navigateur. Si votre connexion s'effectue via un FSI ou derrière un pare-feu à l'aide d'une adresse IP dynamique (via une passerelle NAT depuis un réseau privé), alors votre adresse peut changer périodiquement. Dans ce cas, vous devez trouver la plage d'adresses IP utilisées par les ordinateurs clients :

L'exemple suivant montre comment ajouter une règle pour le protocole RDP (port TCP 3389) à un groupe de sécurité EC2 -VPC avec l'ID `sg-903004f8` correspondant à votre adresse IP.

Pour commencer, trouvez votre adresse IP.

```
$ curl https://checkip.amazonaws.com
x.x.x.x
```

Vous pouvez ensuite ajouter l'adresse IP à votre groupe de sécurité en exécutant la `aws ec2 authorize-security-group-ingress` commande.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port
 3389 --cidr x.x.x.x/x
```

La commande suivante ajoute une autre règle pour activer SSH sur les instances dans le même groupe de sécurité.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port
 22 --cidr x.x.x.x/x
```

Pour afficher les modifications apportées au groupe de sécurité, exécutez la `aws ec2 describe-security-groups` commande.

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8
```

```
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": []
                }
            ],
            "Description": "My security group"
            "IpPermissions": [
                {
                    "ToPort": 22,
                    "IpProtocol": "tcp",
                    "IpRanges": [
                        {
                            "CidrIp": "x.x.x.x/x"
                        }
                    ]
                    "UserIdGroupPairs": [],
                    "FromPort": 22
                }
            ],
            "GroupName": "my-sg",
            "OwnerId": "123456789012",
            "GroupId": "sg-903004f8"
        }
    ]
}
```

Supprimer votre groupe de sécurité

Pour supprimer un groupe de sécurité, exécutez la `aws ec2 delete-security-group` commande.

> **Note**
>
> Vous ne pouvez pas supprimer un groupe de sécurité s'il est actuellement connecté à un environnement.

L'exemple de commande suivant supprime un groupe de sécurité EC2 -VPC.

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

Références

AWS CLI référence :

- aws ec2
- aws ec2 authorize-security-group-ingress
- aws ec2 create-security-group
- aws ec2 delete-security-group
- aws ec2 describe-security-groups

Autre référence :

- Documentation Amazon Elastic Compute Cloud
- Pour consulter le AWS SDK et les exemples de AWS CLI code et y contribuer, consultez le référentiel d'exemples de AWS code sur GitHub.

Lancement, mise en vente et suppression d' EC2instances Amazon dans AWS CLI

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour lancer, répertorier et supprimer des instances Amazon Elastic Compute Cloud (Amazon EC2). Si vous lancez une instance qui n'est pas incluse dans le niveau AWS gratuit, vous êtes facturé après le lancement de l'instance et vous êtes facturé pour la durée d'exécution de l'instance, même si elle reste inactive.

> **Note**
>
> Pour des exemples de commandes supplémentaires, consultez le de référence.

Rubriques

- Prérequis
- Lancer votre instance
- Ajouter un périphérique de stockage en mode bloc à votre instance
- Ajouter une balise à votre instance
- Se connecter à votre instance
- Répertorier vos instances
- Suppression de votre instance
- Références

Prérequis

Pour exécuter les `ec2` commandes décrites dans cette rubrique, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Définissez vos autorisations IAM pour autoriser l' EC2 accès à Amazon. Pour plus d'informations sur les autorisations IAM pour Amazon EC2, consultez les politiques IAM pour Amazon EC2 dans le guide de EC2 l'utilisateur Amazon.
- Créez une paire de clés et un groupe de sécurité.
- Sélectionnez une Amazon Machine Image (AMI) et notez l'ID de l'AMI. Pour plus d'informations, consultez la section Trouver une AMI appropriée dans le guide de EC2 l'utilisateur Amazon.

Lancer votre instance

Pour lancer une EC2 instance Amazon à l'aide de l'AMI que vous avez sélectionnée, utilisez la `aws ec2 run-instances` commande. Vous pouvez lancer l'instance dans un cloud privé virtuel (VPC).

Au départ, votre instance apparait dans l'état `pending`, mais passe à l'état `running` après quelques minutes.

L'exemple suivant montre comment lancer une instance `t2.micro` dans le sous-réseau spécifié d'un VPC. Remplacez les valeurs de paramètre *`italicized`* par vos propres valeurs.

```
$ aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --
key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

```
{
    "OwnerId": "123456789012",
    "ReservationId": "r-5875ca20",
    "Groups": [
        {
            "GroupName": "my-sg",
            "GroupId": "sg-903004f8"
        }
    ],
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": null,
            "Platform": "windows",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2013-07-19T02:42:39.000Z",
            "PrivateIpAddress": "10.0.1.114",
            "ProductCodes": [],
            "VpcId": "vpc-1a2b3c4d",
            "InstanceId": "i-5203422c",
            "ImageId": "ami-173d747e",
            "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
            "KeyName": "MyKeyPair",
            "SecurityGroups": [
                {
                    "GroupName": "my-sg",
                    "GroupId": "sg-903004f8"
                }
            ],
            "ClientToken": null,
            "SubnetId": "subnet-6e7f829e",
            "InstanceType": "t2.micro",
            "NetworkInterfaces": [
                {
                    "Status": "in-use",
                    "SourceDestCheck": true,
                    "VpcId": "vpc-1a2b3c4d",
                    "Description": "Primary network interface",
```

```
                "NetworkInterfaceId": "eni-a7edb1c9",
                "PrivateIpAddresses": [
                    {
                        "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
                        "Primary": true,
                        "PrivateIpAddress": "10.0.1.114"
                    }
                ],
                "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
                "Attachment": {
                    "Status": "attached",
                    "DeviceIndex": 0,
                    "DeleteOnTermination": true,
                    "AttachmentId": "eni-attach-52193138",
                    "AttachTime": "2013-07-19T02:42:39.000Z"
                },
                "Groups": [
                    {
                        "GroupName": "my-sg",
                        "GroupId": "sg-903004f8"
                    }
                ],
                "SubnetId": "subnet-6e7f829e",
                "OwnerId": "123456789012",
                "PrivateIpAddress": "10.0.1.114"
            }
        ],
        "SourceDestCheck": true,
        "Placement": {
            "Tenancy": "default",
            "GroupName": null,
            "AvailabilityZone": "us-west-2b"
        },
        "Hypervisor": "xen",
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/sda1",
                "Ebs": {
                    "Status": "attached",
                    "DeleteOnTermination": true,
                    "VolumeId": "vol-877166c8",
                    "AttachTime": "2013-07-19T02:42:39.000Z"
                }
            }
```

```
            ],
            "Architecture": "x86_64",
            "StateReason": {
                "Message": "pending",
                "Code": "pending"
            },
            "RootDeviceName": "/dev/sda1",
            "VirtualizationType": "hvm",
            "RootDeviceType": "ebs",
            "Tags": [
                {
                    "Value": "MyInstance",
                    "Key": "Name"
                }
            ],
            "AmiLaunchIndex": 0
        }
    ]
}
```

Ajouter un périphérique de stockage en mode bloc à votre instance

Un volume de périphérique racine est associé à chaque instance que vous lancez. Vous pouvez utiliser le mappage des périphériques en mode bloc pour spécifier des volumes Amazon Elastic Block Store (Amazon EBS) supplémentaires ou des volumes de stockage d'instance à associer à une instance lors de son lancement.

Pour ajouter un périphérique de stockage en mode bloc, indiquez l'option `--block-device-mappings` lorsque vous utilisez `run-instances`.

L'exemple de paramètre suivant fournit un volume Amazon EBS standard d'une taille de 20 Go et le mappe à votre instance à l'aide de l'identifiant`/dev/sdf`.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"VolumeSize\":20,
\"DeleteOnTermination\":false}}]"
```

L'exemple suivant ajoute un volume Amazon EBS, mappé vers`/dev/sdf`, sur la base d'un instantané existant. Un instantané représente une image qui est chargée sur le volume pour vous. Lorsque vous spécifiez un instantané, vous n'avez pas besoin de spécifier de volume ; il sera suffisamment grand pour contenir votre image. Cependant, si vous spécifiez une taille, elle doit être supérieure ou égale à la taille de l'instantané.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-a1b2c3d4\"}}]"
```

L'exemple suivant permet d'ajouter deux volumes à votre instance. Le nombre de volumes disponibles pour votre instance varie en fonction de son type d'instance.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"VirtualName\":\"ephemeral0\"},{\"DeviceName\":\"/dev/sdg\",\"VirtualName\":\"ephemeral1\"}]"
```

L'exemple suivant crée le mappage (`/dev/sdj`), mais n'alloue pas de volume à l'instance.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdj\",\"NoDevice\":\"\"}]"
```

Pour plus d'informations, consultez la section Block Device Mapping dans le guide de EC2 l'utilisateur Amazon.

Ajouter une balise à votre instance

Une étiquette est une étiquette que vous attribuez à une AWS ressource. Elle vous permet d'ajouter des métadonnées à vos ressources que vous pouvez utiliser à des fins diverses. Pour plus d'informations, consultez la section Marquage de vos ressources dans le guide de l' EC2 utilisateur Amazon.

L'exemple suivant montre comment ajouter une balise portant le nom de clé « `Name` » et la valeur « `MyInstance` » à l'instance spécifiée, à l'aide de la `aws ec2 create-tags` commande.

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

Se connecter à votre instance

Lorsque votre instance est en cours d'exécution, vous pouvez vous y connecter et l'utiliser comme vous le feriez depuis un ordinateur devant lequel vous êtes assis. Pour plus d'informations, consultez Connect to Your Amazon EC2 Instance dans le guide de EC2 l'utilisateur Amazon.

Répertorier vos instances

Vous pouvez utiliser le AWS CLI pour répertorier vos instances et consulter les informations les concernant. Vous pouvez afficher l'ensemble de vos instances, ou filtrer les résultats en fonction des instances qui vous intéressent.

Les exemples suivants montrent comment utiliser la `aws ec2 describe-instances` commande.

La commande suivante répertorie toutes vos instances.

```
$ aws ec2 describe-instances
```

La commande suivante filtre la liste uniquement pour vos instances `t2.micro` et fournit uniquement en sortie les valeurs `InstanceId` pour chaque correspondance.

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query
 "Reservations[].Instances[].InstanceId"
[
    "i-05e998023d9c69f9a"
]
```

La commande suivante liste toutes vos instances ayant la balise `Name=MyInstance`.

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

La commande suivante répertorie les instances qui ont été lancées à l'aide de l'une des AMIs méthodes suivantes : `ami-x0123456ami-y0123456`, et`ami-z0123456`.

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-
y0123456,ami-z0123456"
```

Suppression de votre instance

Mettre fin à une instance la supprime. Vous ne pouvez pas vous y reconnecter une fois que vous l'avez résiliée.

Dès que l'état de l'instance passe à `shutting-down` ou `terminated`, l'instance ne vous est plus facturée. Si vous souhaitez vous reconnecter à une instance ultérieurement, utilisez stop-instances au lieu de `terminate-instances`. Pour plus d'informations, consultez la section Résiliation de votre instance dans le guide de EC2 l'utilisateur Amazon.

Pour supprimer une instance, vous devez utiliser la commande `aws ec2 terminate-instances` pour la supprimer.

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
```

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-5203422c",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Références

AWS CLI référence :

- aws ec2
- aws ec2 create-tags
- aws ec2 describe-instances
- aws ec2 run-instances
- aws ec2 terminate-instances

Autre référence :

- Documentation Amazon Elastic Compute Cloud
- Pour consulter le AWS SDK et les exemples de AWS CLI code et y contribuer, consultez le référentiel d'exemples de AWS code sur GitHub.

Modifiez le type d' EC2 instance Amazon à l'aide d'un script bash dans le AWS CLI

Cet exemple de script bash pour Amazon EC2 modifie le type d'instance d'une EC2 instance Amazon à l'aide du AWS Command Line Interface ()AWS CLI. Il arrête l'instance si elle est en cours d'exécution, change le type d'instance, puis, sur demande, redémarre l'instance. Les scripts Shell sont des programmes conçus pour être exécutés dans une interface de ligne de commande.

> **Note**
>
> Pour des exemples de commandes supplémentaires, consultez le de référence.

Rubriques

- Avant de commencer
- À propos de cet exemple
- Paramètres
- Dépôt de
- Références

Avant de commencer

Avant de pouvoir exécuter l'un des exemples ci-dessous, vous devez effectuer les opérations suivantes.

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called “Installation/mise à jour” et Authentification et informations d'accès pour AWS CLI.
- Le profil que vous utilisez doit disposer d'autorisations autorisant les AWS opérations effectuées dans les exemples.
- Une EC2 instance Amazon en cours d'exécution dans le compte pour laquelle vous êtes autorisé à arrêter et à modifier. Si vous exécutez le script de test, il lance une instance pour vous, teste la modification du type, puis met fin à l'instance.
- AWS La meilleure pratique consiste à accorder à ce code le privilège minimal, ou uniquement les autorisations requises pour effectuer une tâche. Pour plus d'informations, consultez la section Accorder le moindre privilège dans le AWS guide de l'utilisateur d'Identity and Access Management (IAM).
- Ce code n'a pas été testé dans toutes les AWS régions. Certains AWS services ne sont disponibles que dans certaines régions. Pour plus d'informations, voir Points de terminaison de service et quotas dans le Guide de référence AWS général.
- L'exécution de ce code peut entraîner des frais sur votre AWS compte. Il est de votre responsabilité de vous assurer que toutes les ressources créées par ce script sont supprimées lorsque vous en avez fini avec elles.

À propos de cet exemple

Cet exemple est écrit sous forme de fonction dans le fichier de script shell `change_ec2_instance_type.sh` que vous pouvez utiliser `source` à partir d'un autre script ou de la ligne de commande. Chaque fichier de script contient des commentaires décrivant chacune des fonctions. Une fois que la fonction est en mémoire, vous pouvez l'invoquer depuis la ligne de commande. Par exemple, les commandes suivantes modifient le type de l'instance spécifiée en `t2.nano` :

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

Pour obtenir l'exemple complet et les fichiers de script téléchargeables, consultez Modifier le type d' EC2 instance Amazon dans le référentiel d'exemples de AWS code sur GitHub.

Paramètres

-i - (chaîne) Spécifie l'ID d'instance à modifier.

-t - (string) Spécifie le type d' EC2 instance Amazon vers lequel passer.

-r - (switch) Par défaut, ce paramètre n'est pas défini. S'il `-r` est défini, redémarre l'instance après le changement de type.

-f - (switch) Par défaut, le script invite l'utilisateur à confirmer l'arrêt de l'instance avant de procéder au changement. Si elle `-f` est définie, la fonction n'invite pas l'utilisateur avant d'arrêter l'instance à changer de type

-v - (switch) Par défaut, le script fonctionne silencieusement et affiche le résultat uniquement en cas d'erreur. Si cette `-v` option est définie, la fonction affiche son état tout au long de son fonctionnement.

Dépôt de

`change_ec2_instance_type.sh`

Le fichier de script principal contient la `change_ec2_instance_type()` fonction qui exécute les tâches suivantes :

- Vérifie que l' EC2 instance Amazon spécifiée existe.
- À moins que cette option ne `-f` soit sélectionnée, avertit l'utilisateur avant d'arrêter l'instance.

- Modifie le type d'instance
- Si vous définissez `-r`, redémarre l'instance et confirme qu'elle est en cours d'exécution

Consultez le code correspondant `change_ec2_instance_type.sh` à GitHub.

`test_change_ec2_instance_type.sh`

Le `test_change_ec2_instance_type.sh` script de fichier teste les différents chemins de code de la `change_ec2_instance_type` fonction. Si toutes les étapes du script de test fonctionnent correctement, le script de test supprime toutes les ressources qu'il a créées.

Vous pouvez exécuter le script de test avec les paramètres suivants :

- -v - (switch) Chaque test affiche un pass/failure état au fur et à mesure de son exécution. Par défaut, les tests s'exécutent silencieusement et la sortie inclut uniquement le pass/failure statut global final.
- -i - (switch) Le script s'arrête après chaque test pour vous permettre de parcourir les résultats intermédiaires de chaque étape. Vous permet d'examiner l'état actuel de l'instance à l'aide de la EC2 console Amazon. Le script passe à l'étape suivante une fois que vous avez appuyé sur ENTER à l'invite.

Consultez le code correspondant `test_change_ec2_instance_type.sh` à GitHub.

`awsdocs_general.sh`

Le fichier de script `awsdocs_general.sh` contient les fonctions générales utilisées dans les exemples avancés de AWS CLI.

Consultez le code correspondant `awsdocs_general.sh` à GitHub.

Références

AWS CLI référence :

- `aws ec2`
- `aws ec2 describe-instances`
- `aws ec2 modify-instance-attribute`
- `aws ec2 start-instances`

- `aws ec2 stop-instances`
- `aws ec2 wait instance-running`
- `aws ec2 wait instance-stopped`

Autre référence :

- Documentation Amazon Elastic Compute Cloud
- Pour consulter le AWS SDK et les exemples de AWS CLI code et y contribuer, consultez le référentiel d'exemples de AWS code sur GitHub.

Utilisation d'Amazon S3 Glacier dans AWS CLI

Présentation d'Amazon S3 Glacier
Présentation d'Amazon S3 Glacier

Cette rubrique présente des exemples de AWS CLI commandes qui exécutent des tâches courantes pour S3 Glacier. Les exemples montrent comment utiliser le AWS CLI pour télécharger un fichier volumineux sur S3 Glacier en le divisant en plusieurs parties et en les chargeant depuis la ligne de commande.

Vous pouvez accéder aux fonctionnalités d'Amazon S3 Glacier à l'aide du AWS Command Line Interface (AWS CLI). Pour répertorier les AWS CLI commandes pour S3 Glacier, utilisez la commande suivante.

```
aws glacier help
```

> **Note**
>
> Pour des références de commande et des exemples supplémentaires, voir `aws glacier` la référence des AWS CLI commandes.

Rubriques

- Prérequis

- Création d'un coffre-fort Amazon S3 Glacier
- Préparer un fichier pour le chargement
- Lancer un chargement en plusieurs parties et charger des fichiers
- Procéder au chargement
- Ressources

Prérequis

Pour exécuter les `glacier` commandes, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Ce tutoriel utilise plusieurs outils de ligne de commande qui sont généralement préinstallés sur des systèmes d'exploitation de type Unix, y compris Linux et macOS. Les utilisateurs de Windows peuvent utiliser les mêmes outils en installant Cygwin et en exécutant les commandes à partir du terminal Cygwin. Les commandes et utilitaires natifs de Windows qui exécutent les mêmes fonctions sont signalés lorsqu'ils sont disponibles.

Création d'un coffre-fort Amazon S3 Glacier

Créez un coffre à l'aide de la commande `create-vault`.

```
$ aws glacier create-vault --account-id - --vault-name myvault
{
    "location": "/123456789012/vaults/myvault"
}
```

> **Note**
>
> Toutes les commandes S3 Glacier nécessitent un paramètre d'ID de compte. Utilisez le trait d'union (`--account-id -`) pour utiliser le compte actuel.

Préparer un fichier pour le chargement

Créez un fichier pour le chargement test. Les commandes suivantes créent un fichier nommé *`largefile`* contenant exactement 3 MiB de données aléatoires.

Linux ou macOS

```
$ dd if=/dev/urandom of=largefile bs=3145728 count=1
1+0 records in
1+0 records out
3145728 bytes (3.1 MB) copied, 0.205813 s, 15.3 MB/s
```

dd est un utilitaire qui copie un nombre d'octets d'un fichier d'entrée vers un fichier de sortie. L'exemple précédent utilise le fichier de périphérique système `/dev/urandom` comme source de données aléatoires. `fsutil` exécute une fonction similaire sous Windows.

Windows

```
C:\> fsutil file createnew largefile 3145728
File C:\temp\largefile is created
```

Divisez ensuite le fichier en morceaux de 1 MiB (1 048 576 octets) à l'aide d'un séparateur de fichiers.

```
$ split -b 1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

Lancer un chargement en plusieurs parties et charger des fichiers

Créez un téléchargement partitionné dans Amazon S3 Glacier à l'aide de la `initiate-multipart-upload` commande.

```
$ aws glacier initiate-multipart-upload --account-id - --archive-description "multipart
 upload test" --part-size 1048576 --vault-name myvault
{
    "uploadId": "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ",
    "location": "/123456789012/vaults/myvault/multipart-
uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
}
```

S3 Glacier nécessite la taille de chaque partie en octets (1 MiB dans cet exemple), le nom de votre coffre-fort et un ID de compte pour configurer le téléchargement en plusieurs parties. Le AWS

CLI produit un identifiant de téléchargement lorsque l'opération est terminée. Enregistrez l'ID de chargement dans une variable shell en vue d'une utilisation ultérieure.

Linux ou macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

Utilisez ensuite la commande upload-multipart-part pour télécharger chacune des trois parties.

```
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkaa --range 'bytes
 0-1048575/*' --account-id - --vault-name myvault
{
    "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkab --range 'bytes
 1048576-2097151/*' --account-id - --vault-name myvault
{
    "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkac --range 'bytes
 2097152-3145727/*' --account-id - --vault-name myvault
{
    "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
```

> **Note**
>
> L'exemple précédent utilise le signe dollar ($) pour faire référence au contenu de la variable de terminal UPLOADID sur Linux. Sur la ligne de commande Windows, utilisez un signe de pourcentage (%) de chaque côté du nom de la variable (par exemple,%UPLOADID%).

Vous devez spécifier la plage d'octets de chaque pièce lorsque vous la chargez afin que S3 Glacier puisse la réassembler dans le bon ordre. Chaque partie étant de 1 048 576 octets, la première

occupe les octets 0 à 1 048 575, la deuxième les octets 1 048 576 à 2 097 151 et la troisième les octets 2 097 152 à 3 145 727.

Procéder au chargement

Amazon S3 Glacier nécessite un hachage arborescent du fichier d'origine pour confirmer que tous les éléments chargés sont AWS intacts.

Pour calculer un hachage d'arborescence, vous devez fractionner le fichier en parties de 1 Mio et calculez un hachage SHA-256 binaire de chaque partie. Vous fractionnez ensuite la liste des hachages en paires, combinez les deux hachages binaires dans chaque paire et prenez les hachages des résultats. Répétez ce processus jusqu'à ce qu'il ne reste qu'un seul hachage. Si le un nombre de hachages est impair à un niveau, vous devez le promouvoir au niveau suivant sans le modifier.

L'élément clé pour calculer un hachage d'arborescence correctement lors de l'utilisation des utilitaires de ligne de commande consiste à stocker chaque hachage au format binaire et à convertir en hexadécimal uniquement à la dernière étape. La combinaison ou le hachage de la version hexadécimale d'un hachage dans l'arborescence entraîne un résultat incorrect.

> **Note**
>
> Les utilisateurs de Windows peuvent utiliser la commande `type` à la place de `cat`. OpenSSL est disponible pour Windows à l'adresse OpenSSL.org.

Pour calculer un hachage d'arborescence

1. Fractionnez le fichier d'origine en parties de 1 Mio, si ce n'est déjà fait.

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

2. Calculez et stockez le hachage SHA-256 binaire de chaque segment.

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. Combinez les deux premiers hachages et prenez le hachage binaire du résultat.

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. Combinez le hachage parent des segments `aa` et `ab` au hachage de segment `ac` et hachez le résultat, cette fois en sortie hexadécimale. Stockez le résultat dans une variable shell.

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

Enfin, terminez le chargement avec la commande `complete-multipart-upload`. Cette commande prend la taille du fichier d'origine en octets, la valeur de hachage d'arborescence finale au format hexadécimal, ainsi que votre ID de compte et le nom de coffre.

```
$ aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728 --
upload-id $UPLOADID --account-id - --vault-name myvault
{
    "archiveId": "d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAlLGAAONJAzo5QdP-
N83MKqd96Unspoa5H5lItWX-sK8-QS0ZhwsyGiu9-R-
kwWUyS1dSBlmgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
    "checksum": "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
    "location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAlLGAAONJAzo5QdP-N83MKqd96Unspoa5H5lItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSBlmgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

Vous pouvez également vérifier l'état du coffre avec la commande `describe-vault`.

```
$ aws glacier describe-vault --account-id - --vault-name myvault
{
    "SizeInBytes": 3178496,
    "VaultARN": "arn:aws:glacier:us-west-2:123456789012:vaults/myvault",
    "LastInventoryDate": "2018-12-07T00:26:19.028Z",
    "NumberOfArchives": 1,
    "CreationDate": "2018-12-06T21:23:45.708Z",
    "VaultName": "myvault"
}
```

> **Note**
>
> L'état du coffre est mis à jour une fois par jour. Pour plus d'informations, consultez Utilisation des coffres.

Il est maintenant possible de supprimer sans risque le bloc et les fichiers de hachage que vous avez créés.

```
$ rm chunk* hash*
```

Pour plus d'informations sur les téléchargements partitionnés, consultez les sections Chargement de grandes archives par parties et Calcul des sommes de contrôle dans le guide du développeur Amazon S3 Glacier.

Ressources

AWS CLI référence :

- `aws glacier`
- `aws glacier complete-multipart-upload`
- `aws glacier create-vault`
- `aws glacier describe-vault`
- `aws glacier initiate-multipart-upload`

Référence du service :

- Guide du développeur Amazon S3 Glacier
- Téléchargement partiel d'archives volumineuses dans le guide du développeur Amazon S3 Glacier
- Calculer les checksums dans le guide du développeur Amazon S3 Glacier
- Utilisation des coffres-forts dans le guide du développeur Amazon S3 Glacier

Utilisation d'IAM dans AWS CLI

Une introduction à AWS Identity and Access Management

Présentation de AWS Identity and Access Management

Vous pouvez accéder aux fonctionnalités de AWS Identity and Access Management (IAM) à l'aide du AWS Command Line Interface (AWS CLI). Pour répertorier les AWS CLI commandes pour IAM, utilisez la commande suivante.

```
aws iam help
```

Cette rubrique présente des exemples de AWS CLI commandes qui exécutent des tâches courantes pour IAM.

Avant d'exécuter des commandes, définissez vos informations d'identification par défaut. Pour de plus amples informations, veuillez consulter Configuration des paramètres pour AWS CLI.

Pour plus d'informations sur le service IAM, veuillez consulter le Guide de l'utilisateur AWS Identity and Access Management.

Rubriques

- Création d'utilisateurs et de groupes IAM
- Associer une politique gérée par IAM à un utilisateur
- Définition d'un mot de passe initial pour un utilisateur IAM
- Création d'une clé d'accès pour un utilisateur IAM

Création d'utilisateurs et de groupes IAM

Pour créer un groupe et y ajouter un nouvel utilisateur

1. Utilisez la commande `create-group` pour créer le groupe.

```
$ aws iam create-group --group-name MyIamGroup
{
    "Group": {
        "GroupName": "MyIamGroup",
```

```
        "CreateDate": "2018-12-14T03:03:52.834Z",
        "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
        "Path": "/"
    }
}
```

2. Utilisez la commande create-user pour créer l'utilisateur.

```
$ aws iam create-user --user-name MyUser
{
    "User": {
        "UserName": "MyUser",
        "Path": "/",
        "CreateDate": "2018-12-14T03:13:02.581Z",
        "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

3. Utilisez la commande add-user-to-group pour ajouter l'utilisateur au groupe.

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. Pour vérifier que le groupe MyIamGroup contient le code MyUser, utilisez la commande get-group.

```
$ aws iam get-group --group-name MyIamGroup
{
    "Group": {
        "GroupName": "MyIamGroup",
        "CreateDate": "2018-12-14T03:03:52Z",
        "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
        "Path": "/"
    },
    "Users": [
        {
            "UserName": "MyUser",
            "Path": "/",
            "CreateDate": "2018-12-14T03:13:02Z",
            "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/MyUser"
```

```
        }
    ],
    "IsTruncated": "false"
}
```

Associer une politique gérée par IAM à un utilisateur

La stratégie dans cet exemple fournit à l'utilisateur un « accès utilisateur avancé ».

Pour associer une politique gérée par IAM à un utilisateur

1. Déterminez l'ARN (Amazon Resource Name) de la stratégie à joindre. La commande suivante utilise `list-policies` pour trouver l'ARN de la stratégie avec le nom `PowerUserAccess`. Elle stocke ensuite cet ARN dans une variable d'environnement.

```
$ export POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text)      ~
$ echo $POLICYARN
arn:aws:iam::aws:policy/PowerUserAccess
```

2. Pour attacher la stratégie, utilisez la commande [attach-user-policy](), et référencez la variable d'environnement qui contient l'ARN de stratégie.

```
$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN
```

3. Vérifiez que la stratégie est bien attachée à l'utilisateur en exécutant la commande [list-attached-user-policies]().

```
$ aws iam list-attached-user-policies --user-name MyUser
{
    "AttachedPolicies": [
        {
            "PolicyName": "PowerUserAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
        }
    ]
}
```

Pour plus d'informations, consultez Ressources de gestion des accès. Cette rubrique fournit des liens vers un aperçu des autorisations et des politiques, ainsi que des liens vers des exemples de politiques d'accès à Amazon S3 EC2, Amazon et à d'autres services.

Définition d'un mot de passe initial pour un utilisateur IAM

La commande suivante permet `create-login-profile` de définir un mot de passe initial pour l'utilisateur spécifié. Lorsque l'utilisateur se connecte pour la première fois, il doit modifier le mot de passe afin que seul lui le connaisse.

```
$ aws iam create-login-profile --user-name MyUser --password My!User1Login8P@ssword --
password-reset-required
{
    "LoginProfile": {
        "UserName": "MyUser",
        "CreateDate": "2018-12-14T17:27:18Z",
        "PasswordResetRequired": true
    }
}
```

Vous pouvez utiliser la `update-login-profile` commande pour modifier le mot de passe d'un utilisateur.

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

Création d'une clé d'accès pour un utilisateur IAM

Vous pouvez utiliser la commande pour créer une clé d'accès pour un utilisateur. Une clé d'accès est un ensemble d'informations d'identification de sécurité qui comprend un ID de clé d'accès et une clé secrète.

Un utilisateur ne peut créer que deux clés d'accès à la fois. Si vous essayez de créer un troisième ensemble, la commande renvoie une erreur `LimitExceeded`.

```
$ aws iam create-access-key --user-name MyUser
{
    "AccessKey": {
        "UserName": "MyUser",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "Status": "Active",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
```

```
        "CreateDate": "2018-12-14T17:34:16Z"
    }
}
```

Utilisez la commande pour supprimer une clé d'accès pour un utilisateur. Spécifiez quelle clé d'accès supprimer à l'aide de l'ID de clé d'accès.

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

Utilisation d'Amazon S3 dans AWS CLI

Présentation d'Amazon Simple Storage Service (Amazon S3)
Présentation d'Amazon Simple Storage Service (Amazon S3) - Stockage dans le cloud sur AWS

Vous pouvez accéder aux fonctionnalités d'Amazon Simple Storage Service (Amazon S3) à l'aide AWS Command Line Interface du AWS CLI(). Amazon S3 est un service de stockage d'objets hautement évolutif et durable. Amazon S3 est conçu pour fournir une capacité de stockage pratiquement illimitée, ce qui en fait une solution idéale pour un large éventail de besoins de stockage et de gestion des données.

Amazon S3 vous permet de stocker et de récupérer n'importe quel volume de données, qu'il s'agisse de petits fichiers ou de grands ensembles de données, sous forme d'objets. Chaque objet est stocké dans un conteneur appelé bucket, accessible et géré via AWS Management Console ou par programme via AWS SDKs les outils et AWS CLI.

Y compris le stockage de base, Amazon S3 propose également une gamme de fonctionnalités, notamment la gestion du cycle de vie, le versionnement, l'évolutivité et la sécurité. Elles s'intègrent à d'autres, Services AWS ce qui vous permet de créer des solutions basées sur le cloud qui s'adaptent à vos besoins.

AWS CLI Il fournit deux niveaux de commandes pour accéder à Amazon S3 :

- s3 — Commandes personnalisées de haut niveau spécialement conçues pour simplifier l' AWS CLI exécution de tâches courantes, telles que la création, la manipulation, la suppression et la synchronisation d'objets et de compartiments.
- s3api — Expose un accès direct à toutes les opérations de l'API Amazon S3, ce qui vous permet d'effectuer des opérations avancées.

Sujets abordés dans ce guide :

- À l'aide de commandes de haut niveau (s3) dans AWS CLI
- À l'aide des commandes au niveau de l'API (s3api) dans AWS CLI
- Exemple de script pour le cycle de vie du compartiment Amazon S3 dans AWS CLI

À l'aide de commandes de haut niveau (s3) dans AWS CLI

Cette rubrique décrit certaines des commandes que vous pouvez utiliser pour gérer les buckets et les objets Amazon S3 à l'aide des commandes du AWS CLI. Pour les commandes non abordées dans cette rubrique et pour des exemples de commandes supplémentaires, consultez les commandes dans la AWS CLI référence.

Les `aws s3` commandes de haut niveau simplifient la gestion des objets Amazon S3. Ces commandes vous permettent de gérer le contenu d'Amazon S3 en lui-même et avec les annuaires locaux.

Rubriques

- Prérequis
- Avant de commencer
- Création d’un compartiment
- Répertorier les compartiments et les objets
- Supprimer des compartiments
- Supprimer des objets
- Déplacer des objets
- Copie d’objets
- Synchroniser des objets
- Options fréquemment utilisées pour les commandes s3
- Ressources

Prérequis

Pour exécuter les `s3` commandes, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d’informations, consultez the section called “Installation/mise à jour” et Authentification et informations d'accès pour AWS CLI.

- Le profil que vous utilisez doit disposer d'autorisations autorisant les AWS opérations effectuées dans les exemples.
- Comprenez les termes Amazon S3 suivants :
  - Bucket : dossier Amazon S3 de haut niveau.
  - Préfixe : dossier Amazon S3 dans un compartiment.
  - Objet : tout élément hébergé dans un compartiment Amazon S3.

Avant de commencer

Cette section décrit quelques points à prendre en compte avant d'utiliser des `aws s3` commandes.

Chargement d'objets de grande taille

Lorsque vous utilisez `aws s3` des commandes pour charger des objets volumineux dans un compartiment Amazon S3, le chargement partitionné est AWS CLI automatiquement effectué. Vous ne pouvez pas reprendre un téléchargement qui a échoué en utilisant ces `aws s3` commandes.

Si le téléchargement partitionné échoue en raison d'un délai imparti, ou si vous l'avez annulé manuellement AWS CLI, le téléchargement AWS CLI arrête le téléchargement et nettoie tous les fichiers créés. Ce processus peut prendre plusieurs minutes.

Si le processus de téléchargement ou de nettoyage en plusieurs parties est annulé en raison d'une commande d'arrêt ou d'une défaillance du système, les fichiers créés restent dans le compartiment Amazon S3. Pour nettoyer le téléchargement partitionné, utilisez la commande s3api . abort-multipart-upload

Propriétés des fichiers et balises dans les copies en plusieurs parties

Lorsque vous utilisez la AWS CLI version 1 des commandes de l'`aws s3`espace de noms pour copier un fichier d'un emplacement de compartiment Amazon S3 vers un autre emplacement de compartiment Amazon S3, et que cette opération utilise une copie en plusieurs parties, aucune propriété de fichier de l'objet source n'est copiée vers l'objet de destination.

Par défaut, les commandes de la AWS CLI version 2 de l'`s3`espace de noms qui effectuent des copies partitionnées transfèrent toutes les balises et l'ensemble de propriétés suivant de la copie source vers la copie de destination :`content-type`,`content-language`,`content-encoding`,`content-disposition`, `cache-control`expires, et. `metadata`

Cela peut entraîner des appels d' AWS API supplémentaires vers le point de terminaison Amazon S3 qui n'auraient pas été effectués si vous aviez utilisé AWS CLI la version 1. Peuvent être inclus : `HeadObject`, `GetObjectTagging` et `PutObjectTagging`.

Si vous devez modifier ce comportement par défaut dans les commandes de la AWS CLI version 2, utilisez le `--copy-props` paramètre pour spécifier l'une des options suivantes :

- default — La valeur par défaut. Spécifie que la copie inclut toutes les balises attachées à l'objet source et les propriétés englobées par le paramètre `--metadata-directive` utilisé pour les copies non partitionnées : `content-type`, `content-language`, `content-encoding`, `content-disposition`, `cache-control`, `expires` et `metadata`.
- metadata-directive — Spécifie que la copie inclut uniquement les propriétés incluses dans le `--metadata-directive` paramètre utilisé pour les copies non partitionnées. Aucune balise n'est copiée.
- none — Spécifie que la copie n'inclut aucune des propriétés de l'objet source.

Création d’un compartiment

Utilisez la commande pour créer un bucket. Les noms de compartiment doivent être uniques au niveau mondial (uniques sur l'ensemble d'Amazon S3) et doivent être conformes au DNS.

Les noms de compartiments peuvent contenir des lettres minuscules, des chiffres, des traits d'union et des points. Les noms de compartiment peuvent uniquement commencer et se terminer par une lettre ou un chiffre, et ils ne peuvent pas contenir de point à la suite d'un tiret ou d'un autre point.

Syntaxe

```
$ aws s3 mb <target> [--options]
```

exemples de s3 mb

L'exemple suivant crée le `s3://amzn-s3-demo-bucket` bucket.

```
$ aws s3 mb s3://amzn-s3-demo-bucket
```

Répertorier les compartiments et les objets

Pour répertorier vos compartiments, dossiers ou objets, utilisez la commande. L'utilisation de la commande sans cible ni options permet de répertorier tous les compartiments.

Syntaxe

```
$ aws s3 ls <target> [--options]
```

Pour quelques options courantes à utiliser avec cette commande, ainsi que des exemples, consultezOptions fréquemment utilisées pour les commandes s3. Pour une liste complète des options disponibles, reportez-vous à la référence des AWS CLI commandes.

Exemples de s3 ls

L'exemple suivant répertorie tous vos compartiments Amazon S3.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

La commande suivante répertorie tous les objets et préfixes d'un compartiment. Dans cet exemple de sortie, le préfixe `example/` possède un fichier nommé`MyFile1.txt`.

```
$ aws s3 ls s3://amzn-s3-demo-bucket
                           PRE example/
2018-12-04 19:05:48          3 MyFile1.txt
```

Vous pouvez filtrer la sortie vers un préfixe spécifique en l'incluant dans la commande. La commande suivante répertorie les objets dans *`bucket-name/example/`* (c'est-à-dire les objets *`bucket-name`* filtrés par le préfixe*`example/`*).

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32          3 MyFile1.txt
```

Pour afficher uniquement les compartiments et les objets d'une région spécifique, utilisez les options `--region`

```
$ aws s3 ls --region us-east-2
2018-12-06 18:59:32          3 MyFile1.txt
```

Si vous disposez d'une longue liste de compartiments et d'objets, vous pouvez paginer les résultats à l'aide des options `--max-items` ou`--page-size`. L'`--max-items`option limite le nombre total de

compartiments et d'objets renvoyés lors d'un appel et le `--page-size` nombre d'objets répertoriés sur une page.

```
$ aws s3 ls --max-items 100 --page-size 10
```

Pour plus d'informations sur la pagination, reportez-vous aux sections the section called “--taille de page” etthe section called “--max. d'articles”.

Supprimer des compartiments

Pour supprimer un bucket, utilisez la commande.

Syntaxe

```
$ aws s3 rb <target> [--options]
```

exemples S3 RB

L'exemple suivant supprime le `s3://amzn-s3-demo-bucket` compartiment.

```
$ aws s3 rb s3://amzn-s3-demo-bucket
```

Par défaut, le compartiment doit être vide pour que l'opération aboutisse. Pour supprimer un compartiment qui n'est pas vide, vous devez inclure `--force` cette option. Si vous utilisez un compartiment versionné qui contient des objets précédemment supprimés, mais conservés, cette commande ne vous permet pas de supprimer le compartiment. Vous devez d'abord supprimer tout le contenu.

L'exemple suivant supprime tous les objets et préfixes du compartiment, puis supprime le compartiment.

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

Supprimer des objets

Pour supprimer des objets dans un bucket ou dans votre répertoire local, utilisez la commande.

Syntaxe

```
$ aws s3 rm  <target> [--options]
```

Pour quelques options courantes à utiliser avec cette commande, ainsi que des exemples, consultezOptions fréquemment utilisées pour les commandes s3. Pour une liste complète des options, voir la référence des AWS CLI commandes.

Exemples de RM S3

L'exemple suivant supprime `filename.txt` de. `s3://amzn-s3-demo-bucket/example`

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example/filename.txt
```

L'exemple suivant supprime tous les objets de `s3://amzn-s3-demo-bucket/example` l'`--recursive`option.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example --recursive
```

Déplacer des objets

Utilisez la commande pour déplacer des objets depuis un bucket ou un répertoire local. La `s3 mv` commande copie l'objet ou le fichier source vers la destination spécifiée, puis supprime l'objet ou le fichier source.

Syntaxe

```
$ aws s3 mv <source> <target> [--options]
```

Pour quelques options courantes à utiliser avec cette commande, ainsi que des exemples, consultezOptions fréquemment utilisées pour les commandes s3. Pour une liste complète des options disponibles, reportez-vous à la référence des AWS CLI commandes.

> **Warning**
>
> Si vous utilisez n'importe quel type de point d'accès ARNs ou d'alias de point d'accès dans votre source ou votre destination Amazon S3 URIs, vous devez faire très attention à ce que votre Amazon S3 source et destination se URIs résolvent en différents compartiments sous-jacents. Si les compartiments source et de destination sont identiques, le fichier ou l'objet source peut être déplacé sur lui-même, ce qui peut entraîner la suppression accidentelle de votre fichier ou objet source. Pour vérifier que les compartiments source et de destination ne sont pas identiques, utilisez le `--validate-same-s3-paths` paramètre ou définissez la variable `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` d'`true`environnement sur.

exemples de mv s3

L'exemple suivant déplace tous les objets de `s3://amzn-s3-demo-bucket/example` vers`s3://amzn-s3-demo-bucket/`.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

L'exemple suivant déplace un fichier local de votre répertoire de travail actuel vers le compartiment Amazon S3 à l'aide de la `s3 mv` commande.

```
$ aws s3 mv filename.txt s3://amzn-s3-demo-bucket
```

L'exemple suivant déplace un fichier de votre compartiment Amazon S3 vers votre répertoire de travail actuel, où `./` spécifie votre répertoire de travail actuel.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/filename.txt ./
```

Copie d'objets

Utilisez la commande pour copier des objets depuis un bucket ou un répertoire local.

Syntaxe

```
$ aws s3 cp <source> <target> [--options]
```

Vous pouvez utiliser le paramètre Dash pour le streaming de fichiers vers une entrée standard (`stdin`) ou une sortie standard (`stdout`).

> **Warning**
>
> Si vous l'utilisez PowerShell, le shell peut modifier le codage d'un CRLF ou ajouter un CRLF à une entrée ou à une sortie redirigée, ou à une sortie redirigée.

La `s3 cp` commande utilise la syntaxe suivante pour télécharger un flux de fichiers depuis `stdin` un compartiment spécifié.

Syntaxe

```
$ aws s3 cp - <target> [--options]
```

La `s3 cp` commande utilise la syntaxe suivante pour télécharger un flux de fichiers Amazon S3 pour`stdout`.

Syntaxe

```
$ aws s3 cp <target> [--options] -
```

Pour quelques options courantes à utiliser avec cette commande, ainsi que des exemples, consultezOptions fréquemment utilisées pour les commandes s3. Pour la liste complète des options, voir la référence des AWS CLI commandes.

Exemples **`s3 cp`**

L'exemple suivant copie tous les objets de `s3://amzn-s3-demo-bucket/example` vers`s3://amzn-s3-demo-bucket/`.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

L'exemple suivant copie un fichier local de votre répertoire de travail actuel vers le compartiment Amazon S3 avec la `s3 cp` commande.

```
$ aws s3 cp filename.txt s3://amzn-s3-demo-bucket
```

L'exemple suivant copie un fichier de votre compartiment Amazon S3 vers votre répertoire de travail actuel, où `./` indique votre répertoire de travail actuel.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt ./
```

L'exemple suivant utilise echo pour diffuser le texte « hello world » dans le `s3://bucket-name/filename.txt` fichier.

```
$ echo "hello world" | aws s3 cp - s3://amzn-s3-demo-bucket/filename.txt
```

L'exemple suivant diffuse le `s3://amzn-s3-demo-bucket/filename.txt` fichier vers la console `stdout` et en imprime le contenu.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt -
hello world
```

L'exemple suivant diffuse le contenu de `s3://bucket-name/pre` tostdout, utilise la `bzip2` commande pour compresser les fichiers et télécharge le nouveau fichier compressé nommé `key.bz2` tos`3://bucket-name`.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/pre - | bzip2 --best | aws s3 cp - s3://amzn-s3-demo-bucket/key.bz2
```

Synchroniser des objets

La commande synchronise le contenu d'un compartiment et d'un répertoire, ou le contenu de deux compartiments. Généralement, `s3 sync` copie les fichiers ou objets manquants ou périmés entre la source et la cible. Cependant, vous pouvez également indiquer l'option `--delete` pour supprimer des fichiers ou des objets de la cible qui ne sont pas présents dans la source.

Syntaxe

```
$ aws s3 sync <source> <target> [--options]
```

Pour quelques options courantes à utiliser avec cette commande, ainsi que des exemples, consultezOptions fréquemment utilisées pour les commandes s3. Pour une liste complète des options, voir la référence des AWS CLI commandes.

exemples de synchronisation avec S3

L'exemple suivant synchronise le contenu d'un préfixe Amazon S3 nommé path dans le compartiment nommé amzn-s3-demo-bucket avec le répertoire de travail actuel.

`s3 sync`met à jour tous les fichiers dont la taille ou l'heure de modification sont différentes de celles des fichiers portant le même nom à la destination. La sortie affiche les opérations spécifiques effectuées lors de la synchronisation. Notez que l'opération synchronise de manière récursive le sous-répertoire `MySubdirectory` et son contenu avec. `s3://amzn-s3-demo-bucket/path/MySubdirectory`

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
upload: MyFile2.txt to s3://amzn-s3-demo-bucket/path/MyFile2.txt
```

```
upload: MyFile1.txt to s3://amzn-s3-demo-bucket/path/MyFile1.txt
```

L'exemple suivant, qui prolonge le précédent, montre comment utiliser l'`--delete`option.

```
// Delete local file
$ rm ./MyFile1.txt

// Attempt sync without --delete option - nothing happens
$ aws s3 sync . s3://amzn-s3-demo-bucket/path

// Sync with deletion - object is deleted from bucket
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete
delete: s3://amzn-s3-demo-bucket/path/MyFile1.txt

// Delete object from bucket
$ aws s3 rm s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
delete: s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt

// Sync with deletion - local file is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MySubdirectory\MyFile3.txt

// Sync with Infrequent Access storage class
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --storage-class STANDARD_IA
```

Lorsque vous utilisez l'`--delete`option, les `--include` options `--exclude` et peuvent filtrer les fichiers ou les objets à supprimer au cours d'une `s3 sync` opération. Dans ce cas, la chaîne de paramètre doit spécifier les fichiers à exclure de ou à inclure dans la suppression dans le contexte du répertoire cible ou du compartiment. Vous en trouverez un exemple ci-dessous.

```
Assume local directory and s3://amzn-s3-demo-bucket/path currently in sync and each
 contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt
'''

// Sync with delete, excluding files that match a pattern. MyFile88.txt is deleted,
 while remote MyFile1.txt is not.
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete --exclude "path/MyFile?.txt"
delete: s3://amzn-s3-demo-bucket/path/MyFile88.txt
'''
```

```
// Sync with delete, excluding MyFile2.rtf - local file is NOT deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete --exclude "./MyFile2.rtf"
download: s3://amzn-s3-demo-bucket/path/MyFile1.txt to MyFile1.txt
'''

// Sync with delete, local copy of MyFile2.rtf is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MyFile2.rtf
```

Options fréquemment utilisées pour les commandes s3

Les options suivantes sont fréquemment utilisées pour les commandes décrites dans cette rubrique. Pour une liste complète des options que vous pouvez utiliser sur une commande, consultez la commande spécifique dans le version 2 du guide de référence.

liste acl

`s3 sync`et `s3 cp` vous pouvez utiliser l'`--acl`option. Cela vous permet de définir les autorisations d'accès pour les fichiers copiés sur Amazon S3. L'option `--acl` accepte les valeurs `private`, `public-read` et `public-read-write`. Pour plus d'informations, consultez la section Canned ACL dans le guide de l'utilisateur Amazon S3.

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --acl public-read
```

exclure

Lorsque vous utilisez la `s3 rm` commandes`3 cp`, `s3 mvs3 sync`, ou, ou, vous pouvez filtrer les résultats à l'aide de l'`--include`option `--exclude` ou. L'`--exclude`option définit des règles pour exclure uniquement les objets de la commande, et les options s'appliquent dans l'ordre spécifié. Voici un exemple :

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Exclude all .txt files, resulting in only MyFile2.rtf being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt"

// Exclude all .txt files but include all files with the "MyFile*.txt" format,
 resulting in, MyFile1.txt, MyFile2.rtf, MyFile88.txt being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
 "MyFile*.txt"

// Exclude all .txt files, but include all files with the "MyFile*.txt" format,
 but exclude all files with the "MyFile?.txt" format resulting in, MyFile2.rtf and
 MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
 "MyFile*.txt" --exclude "MyFile?.txt"
```

inclure

Lorsque vous utilisez la `s3 rm` commandes`s3 cp`, `s3 mv`s`s3 sync`, ou, vous pouvez filtrer les résultats à l'aide de l'`--include`option `--exclude` ou. L'`--include`option définit des règles pour inclure uniquement les objets spécifiés pour la commande, et les options s'appliquent dans l'ordre indiqué. Voici un exemple :

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Include all .txt files, resulting in MyFile1.txt and MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt"

// Include all .txt files but exclude all files with the "MyFile*.txt" format,
 resulting in no files being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
 "MyFile*.txt"

// Include all .txt files, but exclude all files with the "MyFile*.txt" format, but
 include all files with the "MyFile?.txt" format resulting in MyFile1.txt being
 copied

$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
 "MyFile*.txt" --include "MyFile?.txt"
```

grant (attribuer)

Les commandes `s3 cp`, `s3 mv` et `s3 sync` incluent une option `--grants` qui peut être utilisée pour accorder des autorisations sur l'objet à des utilisateurs ou groupes spécifiés. Définissez l'`--grants`option sur une liste d'autorisations en utilisant la syntaxe suivante. Remplacez `Permission`, `Grantee_Type` et `Grantee_ID` par vos propres valeurs.

Syntaxe

```
--grants Permission=Grantee_Type=Grantee_ID
         [Permission=Grantee_Type=Grantee_ID ...]
```

Chaque valeur contient les éléments suivants :

- *Permission*— Spécifie les autorisations accordées. Peut être réglé sur `readreadacl`,`writeacl`, ou`full`.
- *Grantee_Type*— Spécifie comment identifier le bénéficiaire. Peut être réglé sur `uriemailaddress`, ou`id`.
- *Grantee_ID*— Spécifie le bénéficiaire en fonction *Grantee_Type* de.
  - `uri`— L'URI du groupe. Pour plus d'informations, voir Qui est un bénéficiaire ?
  - `emailaddress`— L'adresse e-mail du compte.
  - `id`— L'identifiant canonique du compte.

Pour plus d'informations sur le contrôle d'accès Amazon S3, consultez la section Contrôle d'accès.

L'exemple suivant copie un objet dans un compartiment. Il accorde des `read` autorisations sur l'objet à tout le monde, ainsi que `full` des autorisations (`readreadacl`, et`writeacl`) au compte associé à celui-ci`user@example.com`.

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://
acs.amazonaws.com/groups/global/AllUsers full=emailaddress=user@example.com
```

Vous pouvez également spécifier une classe de stockage autre que celle par défaut (`REDUCED_REDUNDANCY`ou`STANDARD_IA`) pour les objets que vous chargez sur Amazon S3. Pour ce faire, utilisez l'option `--storage-class`.

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --storage-class REDUCED_REDUNDANCY
```

recursive

Lorsque vous utilisez cette option, la commande est exécutée sur tous les fichiers ou objets situés dans le répertoire ou le préfixe spécifié. L'exemple suivant supprime l'`s3://amzn-s3-demo-bucket/path`intégralité de son contenu.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/path --recursive
```

Ressources

AWS CLI référence :

- `aws s3`
- `aws s3 cp`
- `aws s3 mb`
- `aws s3 mv`
- `aws s3 ls`
- `aws s3 rb`
- `aws s3 rm`
- `aws s3 sync`

Référence du service :

- Utilisation des compartiments Amazon S3 dans le guide de l'utilisateur Amazon S3
- Utilisation d'objets Amazon S3 dans le guide de l'utilisateur Amazon S3
- Répertorier les clés de manière hiérarchique à l'aide d'un préfixe et d'un délimiteur dans le guide de l'utilisateur Amazon S3
- Abandonnez les téléchargements partitionnés vers un compartiment S3 en utilisant le AWS SDK pour .NET (bas niveau) du guide de l'utilisateur Amazon S3

À l'aide des commandes au niveau de l'API (s3api) dans AWS CLI

Les commandes au niveau de l'API (contenues dans le jeu de `s3api` commandes) fournissent un accès direct à Amazon Simple Storage Service (Amazon S3) APIs et permettent certaines opérations qui ne sont pas exposées dans les commandes de haut niveau. `s3` Ces commandes sont l'équivalent des autres services AWS qui fournissent un accès de niveau API à la fonctionnalité des services. Pour plus d'informations sur les `s3` commandes, voir À l'aide de commandes de haut niveau (s3) dans AWS CLI

Cette rubrique fournit des exemples qui montrent comment utiliser les commandes de niveau inférieur mappées à Amazon S3 APIs. En outre, vous trouverez des exemples pour chaque commande de l'API S3 dans la `s3api` section du version 2 du guide de référence.

Rubriques

- Prérequis
- Appliquer une liste ACL personnalisée
- Configurer une stratégie de journalisation
- Ressources

Prérequis

Pour exécuter les `s3api` commandes, vous devez :

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Le profil que vous utilisez doit disposer d'autorisations autorisant les AWS opérations effectuées dans les exemples.
- Comprenez les termes Amazon S3 suivants :
  - Bucket : dossier Amazon S3 de haut niveau.
  - Préfixe : dossier Amazon S3 dans un compartiment.
  - Objet : tout élément hébergé dans un compartiment Amazon S3.

Appliquer une liste ACL personnalisée

Avec les commandes de haut niveau, vous pouvez utiliser l'`--acl`option permettant d'appliquer des listes de contrôle d'accès prédéfinies (ACLs) aux objets Amazon S3. Mais vous ne pouvez pas utiliser cette commande pour définir l'ensemble du bucket ACLs. Toutefois, vous pouvez le faire à l'aide de la commande au `put-bucket-acl` niveau de l'API.

L'exemple suivant montre comment accorder le contrôle total à deux AWS utilisateurs (user1@example.com et user2@example.com) et l'autorisation de lecture à tout le monde. L'identifiant pour « tout le monde » provient d'un URI spécial que vous passez en tant que paramètre.

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-control
 'emailaddress="user1@example.com",emailaddress="user2@example.com"' --grant-read
 'uri="http://acs.amazonaws.com/groups/global/AllUsers"'
```

Pour plus de détails sur la façon de créer le ACLs, consultez PUT Bucket acl dans le manuel Amazon Simple Storage Service API Reference. Les commandes ACL `s3api` dans l'CLI, telles que `put-bucket-acl`, utilisent la même notation d'argument raccourcie.

Configurer une stratégie de journalisation

La commande API `put-bucket-logging` configure une politique de journalisation des compartiments.

Dans l'exemple suivant, l' AWS utilisateur user@example.com dispose d'un contrôle total sur les fichiers journaux, et tous les utilisateurs ont un accès en lecture à ceux-ci. Notez que la `put-bucket-acl` commande est également requise pour accorder au système de livraison de journaux Amazon S3 (spécifié par une URI) les autorisations nécessaires pour lire et écrire les journaux dans le compartiment.

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-read-acp 'URI="http://
acs.amazonaws.com/groups/s3/LogDelivery"' --grant-write 'URI="http://acs.amazonaws.com/
groups/s3/LogDelivery"'
$ aws s3api put-bucket-logging --bucket amzn-s3-demo-bucket --bucket-logging-status
 file://logging.json
```

Le contenu du `logging.json` fichier indiqué dans la commande précédente est le suivant.

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-bucket",
    "TargetPrefix": "amzn-s3-demo-bucketLogs/",
    "TargetGrants": [
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "Type": "Group",
          "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
        },
        "Permission": "READ"
      }
    ]
  }
}
```

Ressources

AWS CLI référence :

- `aws s3api`
- `aws s3api put-bucket-acl`
- `aws s3api put-bucket-logging`

Référence du service :

- Utilisation des compartiments Amazon S3 dans le guide de l'utilisateur Amazon S3
- Utilisation d'objets Amazon S3 dans le guide de l'utilisateur Amazon S3
- Répertorier les clés de manière hiérarchique à l'aide d'un préfixe et d'un délimiteur dans le guide de l'utilisateur Amazon S3
- Abandonnez les téléchargements partitionnés vers un compartiment S3 en utilisant le AWS SDK pour .NET (bas niveau) du guide de l'utilisateur Amazon S3

Exemple de script pour le cycle de vie du compartiment Amazon S3 dans AWS CLI

Cette rubrique utilise un exemple de script bash pour les opérations de cycle de vie des compartiments Amazon S3 à l'aide de AWS Command Line Interface ()AWS CLI. Cet exemple de script utilise l'ensemble de commandes. Les scripts Shell sont des programmes conçus pour être exécutés dans une interface de ligne de commande.

Rubriques

- Avant de commencer
- À propos de cet exemple
- Dépôt de
- Références

Avant de commencer

Avant de pouvoir exécuter l'un des exemples ci-dessous, vous devez effectuer les opérations suivantes.

- Installation et configuration de l' AWS CLI. Pour plus d'informations, consultez the section called "Installation/mise à jour" et Authentification et informations d'accès pour AWS CLI.
- Le profil que vous utilisez doit disposer d'autorisations autorisant les AWS opérations effectuées dans les exemples.

- Il est AWS recommandé d'accorder à ce code le privilège minimal, ou uniquement les autorisations requises pour effectuer une tâche. Pour plus d'informations, consultez Accorder le privilège le plus faible dans le Guide de l'utilisateur IAM.
- Ce code n'a pas été testé dans toutes les AWS régions. Certains AWS services ne sont disponibles que dans certaines régions. Pour plus d'informations, voir Points de terminaison de service et quotas dans le Guide de référence AWS général.
- L'exécution de ce code peut entraîner des frais sur votre AWS compte. Il est de votre responsabilité de vous assurer que toutes les ressources créées par ce script sont supprimées lorsque vous en avez fini avec elles.

Le service Amazon S3 utilise les termes suivants :

- Bucket : dossier Amazon S3 de haut niveau.
- Préfixe : dossier Amazon S3 dans un compartiment.
- Objet : tout élément hébergé dans un compartiment Amazon S3.

À propos de cet exemple

Cet exemple montre comment interagir avec certaines des opérations de base d'Amazon S3 à l'aide d'un ensemble de fonctions dans des fichiers de script shell. Les fonctions se trouvent dans le fichier de script shell nommé`bucket-operations.sh`. Vous pouvez appeler ces fonctions dans un autre fichier. Chaque fichier de script contient des commentaires décrivant chacune des fonctions.

Pour voir les résultats intermédiaires de chaque étape, exécutez le script avec un `-i` paramètre. Vous pouvez consulter l'état actuel du compartiment ou son contenu à l'aide de la console Amazon S3. Le script passe à l'étape suivante uniquement lorsque vous appuyez sur Entrée à l'invite.

Pour obtenir l'exemple complet et les fichiers de script téléchargeables, consultez Amazon S3 Bucket Lifecycle Operations dans le référentiel d'exemples de AWS code sur GitHub.

Dépôt de

L'exemple contient les fichiers suivants :

bucket-operations.sh

Ce fichier de script principal peut être extrait d'un autre fichier. Il inclut des fonctions qui exécutent les tâches suivantes :

- Création d'un bucket et vérification de son existence
- Copier un fichier de l'ordinateur local vers un bucket
- Copier un fichier d'un emplacement de compartiment vers un autre emplacement de compartiment
- Répertorier le contenu d'un bucket
- Supprimer un fichier d'un bucket
- Suppression d'un compartiment

Consultez le code correspondant `bucket-operations.sh` à GitHub.

test-bucket-operations.sh

Le fichier de script shell `test-bucket-operations.sh` montre comment appeler les fonctions en recherchant le `bucket-operations.sh` fichier et en appelant chacune des fonctions. Après avoir appelé des fonctions, le script de test supprime toutes les ressources qu'il a créées.

Consultez le code correspondant `test-bucket-operations.sh` à GitHub.

awsdocs-general.sh

Le fichier de script `awsdocs-general.sh` contient les fonctions générales utilisées dans les exemples de code avancés pour le AWS CLI.

Consultez le code correspondant `awsdocs-general.sh` à GitHub.

Références

AWS CLI référence :

- `aws s3api`

- `aws s3api create-bucket`
- `aws s3api copy-object`
- `aws s3api delete-bucket`
- `aws s3api delete-object`
- `aws s3api head-bucket`
- `aws s3api list-objects`
- `aws s3api put-object`

Autre référence :

- Utilisation des compartiments Amazon S3 dans le guide de l'utilisateur Amazon S3
- Utilisation d'objets Amazon S3 dans le guide de l'utilisateur Amazon S3
- Pour consulter le AWS SDK et les exemples de AWS CLI code et y contribuer, consultez le référentiel d'exemples de AWS code sur GitHub.

Accès à Amazon SNS dans le AWS CLI

Vous pouvez accéder aux fonctionnalités d'Amazon Simple Notification Service (Amazon SNS) à l'aide AWS Command Line Interface du ().AWS CLI Pour répertorier les AWS CLI commandes pour Amazon SNS, utilisez la commande suivante.

```
aws sns help
```

Avant d'exécuter des commandes, définissez vos informations d'identification par défaut. Pour de plus amples informations, veuillez consulter Configuration des paramètres pour AWS CLI.

Cette rubrique présente des exemples de AWS CLI commandes qui exécutent des tâches courantes pour Amazon SNS.

Rubriques

- Créer une rubrique
- S'abonner à une rubrique
- Publier dans une rubrique
- Annuler l'abonnement à une rubrique
- Supprimer une rubrique

Créer une rubrique

Pour créer une rubrique, utilisez la commande `sns create-topic` et spécifiez le nom que vous souhaitez attribuer à la rubrique.

```
$ aws sns create-topic --name my-topic
{
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

Notez le code `TopicArn` de la réponse, que vous utiliserez plus tard pour publier un message.

S'abonner à une rubrique

Pour vous abonner à une rubrique, utilisez la commande `sns subscribe`.

L'exemple suivant spécifie le protocole `email` et une adresse e-mail pour le code `notification-endpoint`.

```
$ aws sns subscribe --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
protocol email --notification-endpoint saanvi@example.com
{
    "SubscriptionArn": "pending confirmation"
}
```

AWS envoie immédiatement un message de confirmation par e-mail à l'adresse que vous avez spécifiée dans la `subscribe` commande. Le message électronique comporte le texte suivant.

```
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:123456789012:my-topic
To confirm this subscription, click or visit the following link (If this was in error
 no action is necessary):
Confirm subscription
```

Une fois que le destinataire clique sur le lien Confirmer l'abonnement, le navigateur du destinataire affiche un message de notification avec des informations similaires à ce qui suit.

```
Subscription confirmed!

You have subscribed saanvi@example.com to the topic:my-topic.
```

```
Your subscription's id is:
arn:aws:sns:us-west-2:123456789012:my-topic:1328f057-de93-4c15-512e-8bb22EXAMPLE

If it was not your intention to subscribe, click here to unsubscribe.
```

Publier dans une rubrique

Pour envoyer un message à tous les abonnés d'un sujet, utilisez la commande.

L'exemple suivant envoie le message « Hello World ! » à tous les abonnés de la rubrique spécifiée.

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
message "Hello World!"
{
    "MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"
}
```

Dans cet exemple, AWS envoie un e-mail avec le texte « Hello World ! » à`saanvi@example.com`.

Annuler l'abonnement à une rubrique

Pour vous désabonner d'un sujet et ne plus recevoir de messages publiés sur ce sujet, utilisez la commande et spécifiez l'ARN du sujet dont vous souhaitez vous désabonner.

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

Pour vérifier que vous vous êtes bien désinscrit, utilisez la commande pour confirmer que l'ARN n'apparaît plus dans la liste.

```
$ aws sns list-subscriptions
```

Supprimer une rubrique

Pour supprimer un sujet, exécutez la commande.

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

Pour vérifier que AWS le sujet a bien été supprimé, utilisez la commande pour confirmer que le sujet n'apparaît plus dans la liste.

```
$ aws sns list-topics
```

AWS CLI exemples de commandes

Les exemples de code présentés dans cette rubrique vous montrent comment utiliser le AWS Command Line Interface with AWS.

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Certains services contiennent des exemples de catégories supplémentaires qui montrent comment tirer parti des bibliothèques ou des fonctions spécifiques au service.

Services

- Exemples d'ACM utilisant AWS CLI
- Exemples d'API Gateway utilisant AWS CLI
- API Gateway HTTP et exemples WebSocket d'API utilisant AWS CLI
- Exemples d'API de gestion d'API Gateway utilisant AWS CLI
- Exemples d'App Mesh utilisant AWS CLI
- Exemples d'utilisation d'App Runner AWS CLI
- AWS AppConfig exemples utilisant AWS CLI
- Exemples d'Application Auto Scaling utilisant AWS CLI
- Exemples d'Application Discovery Service utilisant AWS CLI
- Exemples de signaux d'application utilisant AWS CLI
- AppRegistry exemples utilisant AWS CLI
- Exemples d'Athena utilisant AWS CLI

- Exemples d'Auto Scaling utilisant AWS CLI
- Exemples d'Auto Scaling Plans utilisant AWS CLI
- AWS Backup exemples utilisant AWS CLI
- AWS Batch exemples utilisant AWS CLI
- AWS Budgets exemples utilisant AWS CLI
- Exemples d'utilisation d'Amazon Chime AWS CLI
- Exemples d'API Cloud Control utilisant AWS CLI
- AWS Cloud Map exemples utilisant AWS CLI
- AWS Cloud9 exemples utilisant AWS CLI
- AWS CloudFormation exemples utilisant AWS CLI
- CloudFront exemples utilisant AWS CLI
- CloudSearch Exemples d'Amazon utilisant AWS CLI
- CloudTrail exemples utilisant AWS CLI
- CloudWatch exemples utilisant AWS CLI
- CloudWatch Exemples de journaux utilisant AWS CLI
- CloudWatch Exemples de surveillance réseau à l'aide de AWS CLI
- CloudWatch Exemples d'Observability Access Monitor utilisant AWS CLI
- CloudWatch Exemples d'administration de l'observabilité utilisant AWS CLI
- CloudWatch Exemples de produits Synthetics utilisant AWS CLI
- CodeArtifact exemples utilisant AWS CLI
- CodeBuild exemples utilisant AWS CLI
- CodeCommit exemples utilisant AWS CLI
- CodeDeploy exemples utilisant AWS CLI
- CodeGuru Exemples de réviseurs utilisant AWS CLI
- CodePipeline exemples utilisant AWS CLI
- AWS CodeStar Exemples de notifications utilisant AWS CLI
- CodeConnections exemples utilisant AWS CLI
- Exemples d'Amazon Cognito Identity utilisant AWS CLI
- Exemples d'utilisation du fournisseur d'identité Amazon Cognito AWS CLI
- Exemples d'Amazon Comprehend utilisant AWS CLI

- Exemples d'Amazon Comprehend Medical utilisant AWS CLI
- AWS Config exemples utilisant AWS CLI
- Exemples d'Amazon Connect utilisant AWS CLI
- AWS Cost and Usage Report exemples utilisant AWS CLI
- Exemples du service Cost Explorer utilisant AWS CLI
- Exemples de Firehose utilisant AWS CLI
- Exemples d'Amazon Data Lifecycle Manager utilisant AWS CLI
- AWS Data Pipeline exemples utilisant AWS CLI
- DataSync exemples utilisant AWS CLI
- Exemples DAX utilisant AWS CLI
- Exemples de Detective utilisant AWS CLI
- Exemples d'utilisation de Device Farm AWS CLI
- AWS Direct Connect exemples utilisant AWS CLI
- AWS Directory Service exemples utilisant AWS CLI
- AWS Directory Service Exemples de données utilisant AWS CLI
- AWS DMS exemples utilisant AWS CLI
- Exemples d'Amazon DocumentDB utilisant AWS CLI
- Exemples DynamoDB utilisant AWS CLI
- Exemples de DynamoDB Streams utilisant AWS CLI
- EC2 Exemples Amazon utilisant AWS CLI
- Exemples d'Amazon EC2 Instance Connect utilisant AWS CLI
- Exemples d'Amazon ECR utilisant AWS CLI
- Exemples d'Amazon ECR Public utilisant AWS CLI
- Exemples d'Amazon ECS utilisant AWS CLI
- Exemples d'Amazon EFS utilisant AWS CLI
- Exemples d'utilisation d'Amazon EKS AWS CLI
- Exemples d'utilisation d'Elastic Beanstalk AWS CLI
- Elastic Load Balancing - Version 1 : exemples d'utilisation AWS CLI
- Elastic Load Balancing - Version 2 : exemples d'utilisation AWS CLI
- Exemples d'utilisation d'Elastic Transcoder AWS CLI

- ElastiCache exemples utilisant AWS CLI
- MediaStore exemples utilisant AWS CLI
- Exemples d'Amazon EMR utilisant AWS CLI
- Exemples d'Amazon EMR sur EKS utilisant AWS CLI
- EventBridge exemples utilisant AWS CLI
- EventBridge Exemples de tuyaux utilisant AWS CLI
- Exemples d'utilisation de Firewall Manager AWS CLI
- AWS FIS exemples utilisant AWS CLI
- Exemples d'utilisation de Amazon GameLift Servers AWS CLI
- Exemples de Global Accelerator utilisant AWS CLI
- AWS Glue exemples utilisant AWS CLI
- GuardDuty exemples utilisant AWS CLI
- AWS Health exemples utilisant AWS CLI
- HealthImaging exemples utilisant AWS CLI
- HealthLake exemples utilisant AWS CLI
- HealthOmics exemples utilisant AWS CLI
- Exemples d'IAM utilisant AWS CLI
- Exemples d'utilisation d'IAM Access Analyzer AWS CLI
- Exemples d'Image Builder utilisant AWS CLI
- Exemples d'utilisation d'Incident Manager AWS CLI
- Exemples de contacts du gestionnaire d'incidents utilisant AWS CLI
- Exemples d'utilisation d'Amazon Inspector AWS CLI
- AWS IoT exemples utilisant AWS CLI
- AWS IoT Analytics exemples utilisant AWS CLI
- Exemples d'utilisation de Device Advisor AWS CLI
- AWS IoT data exemples utilisant AWS CLI
- AWS IoT Events exemples utilisant AWS CLI
- AWS IoT Events-Data exemples utilisant AWS CLI
- AWS IoT Greengrass exemples utilisant AWS CLI
- AWS IoT Greengrass V2 exemples utilisant AWS CLI

- AWS IoT Jobs SDK release exemples utilisant AWS CLI
- AWS IoT SiteWise exemples utilisant AWS CLI
- AWS IoT Things Graph exemples utilisant AWS CLI
- AWS IoT Wireless exemples utilisant AWS CLI
- Exemples d'utilisation d'Amazon IVS AWS CLI
- Exemples d'Amazon IVS Chat utilisant AWS CLI
- Exemples de diffusion en temps réel d'Amazon IVS à l'aide d'Amazon IVS AWS CLI
- Exemples d'utilisation d'Amazon Kendra AWS CLI
- Exemples d'utilisation de Kinesis AWS CLI
- AWS KMS exemples utilisant AWS CLI
- Exemples de Lake Formation utilisant AWS CLI
- Exemples Lambda utilisant AWS CLI
- Exemples de License Manager utilisant AWS CLI
- Exemples d'utilisation de Lightsail AWS CLI
- Exemples d'utilisation de Macie AWS CLI
- Exemples d'Amazon Managed Grafana utilisant AWS CLI
- MediaConnect exemples utilisant AWS CLI
- MediaConvert exemples utilisant AWS CLI
- MediaLive exemples utilisant AWS CLI
- MediaPackage exemples utilisant AWS CLI
- MediaPackage Exemples de VOD utilisant AWS CLI
- MediaStore Exemples de plans de données utilisant AWS CLI
- MediaTailor exemples utilisant AWS CLI
- Exemples de MemoryDB utilisant AWS CLI
- Exemples d'Amazon MSK utilisant AWS CLI
- Exemples d'utilisation de Network Flow Monitor AWS CLI
- Exemples d'utilisation de Network Manager AWS CLI
- OpenSearch Exemples de services utilisant AWS CLI
- Organisations utilisant des exemples AWS CLI
- AWS Outposts exemples utilisant AWS CLI

- AWS Payment Cryptography exemples utilisant AWS CLI
- AWS Payment Cryptography Exemples de plans de données utilisant AWS CLI
- Exemples d'utilisation d'Amazon Pinpoint AWS CLI
- Exemples d'utilisation d'Amazon Polly AWS CLI
- AWS Price List exemples utilisant AWS CLI
- AWS Private CA exemples utilisant AWS CLI
- AWS Proton exemples utilisant AWS CLI
- Exemples de QLDB utilisant AWS CLI
- Exemples d'Amazon RDS utilisant AWS CLI
- Exemples d'Amazon RDS Data Service utilisant AWS CLI
- Exemples d'Amazon RDS Performance Insights utilisant AWS CLI
- Exemples d'utilisation d'Amazon Redshift AWS CLI
- Exemples d'utilisation d'Amazon Rekognition AWS CLI
- AWS RAM exemples utilisant AWS CLI
- Exemples d'utilisation de Resource Explorer AWS CLI
- Exemples de Resource Groups utilisant AWS CLI
- Exemples d'API de balisage Resource Groups utilisant AWS CLI
- AWS RoboMaker exemples utilisant AWS CLI
- Exemples de Route 53 utilisant AWS CLI
- Exemples d'enregistrement de domaine Route 53 à l'aide de AWS CLI
- Exemples de profils Route 53 utilisant AWS CLI
- Exemples de résolveurs Route 53 utilisant AWS CLI
- Exemples d'utilisation d'Amazon S3 AWS CLI
- Exemples d'Amazon S3 Control utilisant AWS CLI
- Exemples d'utilisation de S3 Glacier AWS CLI
- Exemples d'utilisation de Secrets Manager AWS CLI
- Exemples d'utilisation de Security Hub AWS CLI
- Exemples d'utilisation de Security Lake AWS CLI
- AWS Serverless Application Repository exemples utilisant AWS CLI
- Exemples de Service Catalog utilisant AWS CLI

- Exemples de quotas de service utilisant AWS CLI
- Exemples d'utilisation d'Amazon SES AWS CLI
- Exemples de Shield utilisant AWS CLI
- Exemples de signataires utilisant AWS CLI
- Exemples d'utilisation de Snowball Edge AWS CLI
- Exemples d'utilisation d'Amazon SNS AWS CLI
- Exemples d'utilisation d'Amazon SQS AWS CLI
- Exemples de Storage Gateway utilisant AWS CLI
- AWS STS exemples utilisant AWS CLI
- Support exemples utilisant AWS CLI
- Exemples d'Amazon SWF utilisant AWS CLI
- Exemples de Systems Manager utilisant AWS CLI
- Exemples d'Amazon Textract utilisant AWS CLI
- Exemples d'Amazon Transcribe utilisant AWS CLI
- Exemples d'Amazon Translate utilisant AWS CLI
- Trusted Advisor exemples utilisant AWS CLI
- Exemples d'autorisations vérifiées utilisant AWS CLI
- Exemples de treillis VPC utilisant AWS CLI
- AWS WAF Classic exemples utilisant AWS CLI
- AWS WAF Classic Regional exemples utilisant AWS CLI
- AWS WAFV2 exemples utilisant AWS CLI
- WorkDocs exemples utilisant AWS CLI
- WorkMail Exemples d'Amazon utilisant AWS CLI
- Exemples d'Amazon WorkMail Message Flow utilisant AWS CLI
- WorkSpaces exemples utilisant AWS CLI
- Exemples de X-Ray utilisant AWS CLI

Exemples d'ACM utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with ACM.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-certificate`

L'exemple de code suivant montre comment utiliser`add-tags-to-certificate`.

AWS CLI

Pour ajouter des balises à un certificat ACM existant

La `add-tags-to-certificate` commande suivante ajoute deux balises au certificat spécifié. Utilisez un espace pour séparer plusieurs balises :

```
aws acm add-tags-to-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- Pour plus de détails sur l'API, reportez-vous AddTagsToCertificateà la section Référence des AWS CLI commandes.

`delete-certificate`

L'exemple de code suivant montre comment utiliser`delete-certificate`.

AWS CLI

Pour supprimer un certificat ACM de votre compte

La `delete-certificate` commande suivante supprime le certificat avec l'ARN spécifié :

```
aws acm delete-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- Pour plus de détails sur l'API, reportez-vous DeleteCertificateà la section Référence des AWS CLI commandes.

`describe-certificate`

L'exemple de code suivant montre comment utiliser`describe-certificate`.

AWS CLI

Pour récupérer les champs contenus dans un certificat ACM

La `describe-certificate` commande suivante extrait tous les champs du certificat avec l'ARN spécifié :

```
aws acm describe-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Un résultat similaire à ce qui suit s'affiche :

```
{
  "Certificate": {
    "CertificateArn":
 "arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
    "CreatedAt": 1446835267.0,
    "DomainName": "www.example.com",
    "DomainValidationOptions": [
      {
        "DomainName": "www.example.com",
        "ValidationDomain": "www.example.com",
        "ValidationEmails": [
          "hostmaster@example.com",
          "admin@example.com",
          "owner@example.com.whoisprivacyservice.org",
          "tech@example.com.whoisprivacyservice.org",
          "admin@example.com.whoisprivacyservice.org",
          "postmaster@example.com",
          "webmaster@example.com",
          "administrator@example.com"
        ]
```

```
      },
      {
        "DomainName": "www.example.net",
        "ValidationDomain": "www.example.net",
        "ValidationEmails": [
          "postmaster@example.net",
          "admin@example.net",
          "owner@example.net.whoisprivacyservice.org",
          "tech@example.net.whoisprivacyservice.org",
          "admin@example.net.whoisprivacyservice.org",
          "hostmaster@example.net",
          "administrator@example.net",
          "webmaster@example.net"
        ]
      }
    ],
    "InUseBy": [],
    "IssuedAt": 1446835815.0,
    "Issuer": "Amazon",
    "KeyAlgorithm": "RSA-2048",
    "NotAfter": 1478433600.0,
    "NotBefore": 1446768000.0,
    "Serial": "0f:ac:b0:a3:8d:ea:65:52:2d:7d:01:3a:39:36:db:d6",
    "SignatureAlgorithm": "SHA256WITHRSA",
    "Status": "ISSUED",
    "Subject": "CN=www.example.com",
    "SubjectAlternativeNames": [
      "www.example.com",
      "www.example.net"
    ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeCertificateà la section Référence des AWS CLI commandes.

`export-certificate`

L'exemple de code suivant montre comment utiliser`export-certificate`.

AWS CLI

Pour exporter un certificat privé émis par une autorité de certification privée.

La `export-certificate` commande suivante exporte un certificat privé, une chaîne de certificats et une clé privée vers votre écran :

```
aws acm export-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file
```

Pour exporter le certificat, la chaîne et la clé privée vers un fichier local, utilisez la commande suivante :

```
aws acm export-certificate --certificate-
arn arn:aws:acm:region:sccount:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file > c:\temp\export.txt
```

- Pour plus de détails sur l'API, reportez-vous ExportCertificateà la section Référence des AWS CLI commandes.

`get-certificate`

L'exemple de code suivant montre comment utiliser`get-certificate`.

AWS CLI

Pour récupérer un certificat ACM

La `get-certificate` commande suivante permet de récupérer le certificat pour l'ARN spécifié et la chaîne de certificats :

```
aws acm get-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Un résultat similaire à ce qui suit s'affiche :

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
```

```
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",

  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
```

```
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
}
```

- Pour plus de détails sur l'API, reportez-vous GetCertificateà la section Référence des AWS CLI commandes.

`import-certificate`

L'exemple de code suivant montre comment utiliser`import-certificate`.

AWS CLI

Pour importer un certificat dans ACM.

La `import-certificate` commande suivante importe un certificat dans ACM. Remplacez les noms de fichiers par les vôtres :

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-
chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- Pour plus de détails sur l'API, reportez-vous ImportCertificateà la section Référence des AWS CLI commandes.

`list-certificates`

L'exemple de code suivant montre comment utiliser`list-certificates`.

AWS CLI

Pour répertorier les certificats ACM d'un AWS compte

La `list-certificates` commande suivante répertorie les certificats ARNs de votre compte :

```
aws acm list-certificates
```

La commande précédente génère une sortie similaire à la sortie suivante :

```
{
    "CertificateSummaryList": [
        {
            "CertificateArn":
 "arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
            "DomainName": "www.example.com"
        },
        {
            "CertificateArn": "arn:aws:acm:region:account:certificate/aaaaaaaa-bbbb-
cccc-dddd-eeeeeeeeeeee",
            "DomainName": "www.example.net"
        }
    ]
}
```

Vous pouvez décider du nombre de certificats que vous souhaitez afficher à chaque appel`list-certificates`. Par exemple, si vous avez quatre certificats et que vous ne souhaitez pas en afficher plus de deux à la fois, définissez l'`max-items`argument sur 2 comme dans l'exemple suivant :

```
aws acm list-certificates --max-items 2
```

Deux certificats ARNs et une `NextToken` valeur seront affichés :

```
"CertificateSummaryList": [
  {
    "CertificateArn": "arn:aws:acm:region:account: \
            certificate/12345678-1234-1234-1234-123456789012",
    "DomainName": "www.example.com"
  },
  {
```

```
    "CertificateArn": "arn:aws:acm:region:account: \
             certificate/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "DomainName": "www.example.net"
  }
  ],
    "NextToken": "9f4d9f69-275a-41fe-b58e-2b837bd9ba48"
```

Pour afficher les deux certificats suivants dans votre compte, définissez cette `NextToken` valeur lors de votre prochain appel :

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-
b58e-2b837bd9ba48
```

Vous pouvez filtrer votre sortie à l'aide de l'`certificate-statuses`argument. La commande suivante affiche les certificats dotés du statut PENDING_VALIDATION :

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

Vous pouvez également filtrer votre sortie à l'aide de l'`includes`argument. La commande suivante affiche les certificats filtrés selon les propriétés suivantes. Les certificats à afficher :

```
- Specify that the RSA algorithm and a 2048 bit key are used to generate key pairs.
- Contain a Key Usage extension that specifies that the certificates can be used to
 create digital signatures.
- Contain an Extended Key Usage extension that specifies that the certificates can
 be used for code signing.

aws acm list-certificates --max-items 10 --includes
 extendedKeyUsage=CODE_SIGNING,keyUsage=DIGITAL_SIGNATURE,keyTypes=RSA_2048
```

- Pour plus de détails sur l'API, reportez-vous ListCertificatesà la section Référence des AWS CLI commandes.

`list-tags-for-certificate`

L'exemple de code suivant montre comment utiliser`list-tags-for-certificate`.

AWS CLI

Pour répertorier les balises appliquées à un certificat ACM

La `list-tags-for-certificate` commande suivante répertorie les balises appliquées à un certificat dans votre compte :

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

La commande précédente génère une sortie similaire à la sortie suivante :

```
{
  "Tags": [
      {
          "Value": "Website",
          "Key": "Purpose"
      },
      {
          "Value": "Alice",
          "Key": "Admin"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForCertificateà la section Référence des AWS CLI commandes.

`remove-tags-from-certificate`

L'exemple de code suivant montre comment utiliser`remove-tags-from-certificate`.

AWS CLI

Pour supprimer une étiquette d'un certificat ACM

La `remove-tags-from-certificate` commande suivante supprime deux balises du certificat spécifié. Utilisez un espace pour séparer plusieurs balises :

```
aws acm remove-tags-from-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- Pour plus de détails sur l'API, reportez-vous RemoveTagsFromCertificateà la section Référence des AWS CLI commandes.

`request-certificate`

L'exemple de code suivant montre comment utiliser`request-certificate`.

AWS CLI

Pour demander un nouveau certificat ACM

La `request-certificate` commande suivante demande un nouveau certificat pour le domaine www.example.com à l'aide de la validation DNS :

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

Vous pouvez saisir un jeton d'idempuissance pour faire la distinction entre les appels à : `request-certificate`

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --
idempotency-token 91adc45q
```

Vous pouvez saisir un ou plusieurs noms de sujet alternatifs pour demander un certificat qui protégera plusieurs domaines apex :

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names www.example.net
```

Vous pouvez saisir un autre nom qui peut également être utilisé pour accéder à votre site Web :

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

Vous pouvez utiliser un astérisque (*) comme caractère générique pour créer un certificat pour plusieurs sous-domaines du même domaine :

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

Vous pouvez également saisir plusieurs noms alternatifs :

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
subject-alternative-names b.example.com c.example.com d.example.com
```

Si vous utilisez le courrier électronique pour la validation, vous pouvez saisir des options de validation de domaine pour spécifier le domaine auquel l'e-mail de validation sera envoyé :

```
aws acm request-certificate --domain-name example.com --validation-
method EMAIL --subject-alternative-names www.example.com --domain-validation-
options DomainName=example.com,ValidationDomain=example.com
```

La commande suivante permet de désactiver la journalisation transparente des certificats lorsque vous demandez un nouveau certificat :

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- Pour plus de détails sur l'API, reportez-vous RequestCertificateà la section Référence des AWS CLI commandes.

`resend-validation-email`

L'exemple de code suivant montre comment utiliser`resend-validation-email`.

AWS CLI

Pour renvoyer un e-mail de validation pour votre demande de certificat ACM

La `resend-validation-email` commande suivante indique à l'autorité de certification Amazon d'envoyer un e-mail de validation aux adresses appropriées :

```
aws acm resend-validation-email --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
domain www.example.com --validation-domain example.com
```

- Pour plus de détails sur l'API, reportez-vous ResendValidationEmailà la section Référence des AWS CLI commandes.

`update-certificate-options`

L'exemple de code suivant montre comment utiliser`update-certificate-options`.

AWS CLI

Pour mettre à jour les options de certificat

La `update-certificate-options` commande suivante permet de désactiver la journalisation de la transparence des certificats :

```
aws acm update-certificate-options --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
options CertificateTransparencyLoggingPreference=DISABLED
```

- Pour plus de détails sur l'API, reportez-vous UpdateCertificateOptionsà la section Référence des AWS CLI commandes.

Exemples d'API Gateway utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with API Gateway.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-api-key`

L'exemple de code suivant montre comment utiliser`create-api-key`.

AWS CLI

Pour créer une clé d'API activée pour une API et un stage existants

Commande :

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for
 development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- Pour plus de détails sur l'API, reportez-vous CreateApiKeyà la section Référence des AWS CLI commandes.

`create-authorizer`

L'exemple de code suivant montre comment utiliser`create-authorizer`.

AWS CLI

Exemple 1 : pour créer un autorisateur personnalisé API Gateway basé sur des jetons pour l'API

L'`create-authorizer`exemple suivant crée un système d'autorisation basé sur des jetons.

```
aws apigateway create-authorizer \
    --rest-api-id 1234123412 \
    --name 'First_Token_Custom_Authorizer' \
    --type TOKEN \
    --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
    --identity-source 'method.request.header.Authorization' \
    --authorizer-result-ttl-in-seconds 300
```

Sortie :

```
{
    "authType": "custom",
    "name": "First_Token_Custom_Authorizer",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
    "authorizerResultTtlInSeconds": 300,
    "identitySource": "method.request.header.Authorization",
    "type": "TOKEN",
    "id": "z40xj0"
}
```

Exemple 2 : pour créer un autorisateur personnalisé API Gateway basé sur les groupes d'utilisateurs Cognito pour l'API

L'`create-authorizer`exemple suivant crée un autorisateur personnalisé API Gateway basé sur les groupes d'utilisateurs Cognito.

```
aws apigateway create-authorizer \
```

```
    --rest-api-id 1234123412 \
    --name 'First_Cognito_Custom_Authorizer' \
    --type COGNITO_USER_POOLS \
    --provider-arns 'arn:aws:cognito-idp:us-east-1:123412341234:userpool/us-
east-1_aWcZeQbuD' \
    --identity-source 'method.request.header.Authorization'
```

Sortie :

```
{
    "authType": "cognito_user_pools",
    "identitySource": "method.request.header.Authorization",
    "name": "First_Cognito_Custom_Authorizer",
    "providerARNs": [
        "arn:aws:cognito-idp:us-east-1:342398297714:userpool/us-east-1_qWbZzQhzE"
    ],
    "type": "COGNITO_USER_POOLS",
    "id": "5yid1t"
}
```

Exemple 3 : pour créer un autorisateur personnalisé API Gateway basé sur les demandes pour l'API

L'`create-authorizer`exemple suivant crée un mécanisme d'autorisation basé sur les demandes.

```
aws apigateway create-authorizer \
    --rest-api-id 1234123412 \
    --name 'First_Request_Custom_Authorizer' \
    --type REQUEST \
    --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
    --identity-source 'method.request.header.Authorization,context.accountId' \
    --authorizer-result-ttl-in-seconds 300
```

Sortie :

```
{
    "id": "z40xj0",
    "name": "First_Request_Custom_Authorizer",
    "type": "REQUEST",
```

```
    "authType": "custom",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
    "identitySource": "method.request.header.Authorization,context.accountId",
    "authorizerResultTtlInSeconds": 300
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAuthorizerà la section Référence des AWS CLI commandes.

`create-base-path-mapping`

L'exemple de code suivant montre comment utiliser`create-base-path-mapping`.

AWS CLI

Pour créer le mappage de chemin de base pour un nom de domaine personnalisé

Commande :

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-
api-id 1234123412 --stage prod --base-path v1
```

- Pour plus de détails sur l'API, reportez-vous CreateBasePathMappingà la section Référence des AWS CLI commandes.

`create-deployment`

L'exemple de code suivant montre comment utiliser`create-deployment`.

AWS CLI

Pour déployer les ressources configurées pour une API vers une nouvelle étape

Commande :

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-
description 'Development Stage' --description 'First deployment to the dev stage'
```

Pour déployer les ressources configurées pour une API sur une étape existante

Commande :

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --
description 'Second deployment to the dev stage'
```

Pour déployer les ressources configurées pour une API sur une étape existante à l'aide de variables d'étape

aws apigateway create-deployment -- rest-api-id 1234123412 --stage-name dev -- description 'Troisième déploiement vers la phase de développement' --variables key='value', otherKey='othervalue'

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentà la section Référence des AWS CLI commandes.

`create-domain-name-access-association`

L'exemple de code suivant montre comment utiliser`create-domain-name-access-association`.

AWS CLI

Pour créer une association d'accès à un nom de domaine

L'`create-domain-name-access-association`exemple suivant crée une association d'accès au nom de domaine entre un nom de domaine personnalisé privé et un point de terminaison VPC.

```
aws apigateway create-domain-name-access-association \
    --domain-name-arn arn:aws:apigateway:us-west-2:111122223333:/domainnames/
my.private.domain.tld+abcd1234 \
    --access-association-source vpce-abcd1234efg \
    --access-association-source-type VPCE
```

Sortie :

```
{
    "domainNameAccessAssociationArn": "arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg
    "accessAssociationSource": "vpce-abcd1234efg",
```

```
    "accessAssociationSourceType": "VPCE",
    "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
}
```

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDomainNameAccessAssociationà la section Référence des AWS CLI commandes.

`create-domain-name`

L'exemple de code suivant montre comment utiliser`create-domain-name`.

AWS CLI

Exemple 1 : pour créer un nom de domaine public personnalisé

L'`create-domain-name`exemple suivant crée un nom de domaine public personnalisé.

```
aws apigateway create-domain-name \
    --domain-name 'my.domain.tld' \
    --certificate-name 'my.domain.tld cert'\
    --certificate-arn 'arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3'
```

Sortie :

```
{
    "domainName": "my.domain.tld",
    "certificateName": "my.domain.tld cert",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-10-08T11:29:49-07:00",
    "distributionDomainName": "abcd1234.cloudfront.net",
    "distributionHostedZoneId": "Z2FDTNDATAQYW2",
    "endpointConfiguration": {
        "types": [
            "EDGE"
        ]
    },
```

```
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2"
}
```

Pour plus d'informations, consultez la section Nom de domaine personnalisé pour le REST public APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Exemple 2 : pour créer un nom de domaine personnalisé privé

L'`create-domain-name`exemple suivant crée un nom de domaine personnalisé privé.

```
aws apigateway create-domain-name \
    --domain-name 'my.private.domain.tld' \
    --certificate-name 'my.domain.tld cert' \
    --certificate-arn 'arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3' \
    --endpoint-configuration '{"types": ["PRIVATE"]}' \
    --security-policy 'TLS_1_2' \
    --policy file://policy.json
```

Contenu de `policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "execute-api:Invoke",
            "Resource": [
                "execute-api:/*"
            ]
        },
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "execute-api:Invoke",
            "Resource": [
                "execute-api:/*"
            ],
            "Condition" : {
                "StringNotEquals": {
```

```
                    "aws:SourceVpce": "vpce-abcd1234efg"
                }
            }
        }
    ]
}
```

Sortie :

```
{
    "domainName": "my.private.domain.tld",
    "domainNameId": "abcd1234",
    "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-09-10T10:31:20-07:00",
    "endpointConfiguration": {
        "types": [
            "PRIVATE"
        ]
    },
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":\"*\",\"Action\":\"execute-api:Invoke\",\"Resource\":
\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/my.private.domain.tld
+abcd1234\"},{\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"execute-
api:Invoke\",\"Resource\":\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234\",\"Condition\":{\"StringNotEquals\":{\"aws:SourceVpc
\":\"vpc-1a2b3c4d\"}}}]}"
}
```

Pour plus d'informations, consultez la section Nom de domaine personnalisé pour le REST public APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDomainNameà la section Référence des AWS CLI commandes.

`create-model`

L'exemple de code suivant montre comment utiliser`create-model`.

AWS CLI

Pour créer un modèle pour une API

Commande :

```
aws apigateway create-model --rest-api-id 1234123412 --name 'firstModel' --
description 'The First Model' --content-type 'application/json'  --schema
 '{ "$schema": "http://json-schema.org/draft-04/schema#", "title": "firstModel",
 "type": "object", "properties": { "firstProperty" : { "type": "object",
 "properties": { "key": { "type": "string" } } } } }'
```

Sortie :

```
{
    "contentType": "application/json",
    "description": "The First Model",
    "name": "firstModel",
    "id": "2rzg0l",
    "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title
\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty
\" : { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string
\" } } } } }"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateModelà la section Référence des AWS CLI commandes.

`create-resource`

L'exemple de code suivant montre comment utiliser`create-resource`.

AWS CLI

Pour créer une ressource dans une API

Commande :

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-
part 'new-resource'
```

- Pour plus de détails sur l'API, reportez-vous CreateResourceà la section Référence des AWS CLI commandes.

`create-rest-api`

L'exemple de code suivant montre comment utiliser`create-rest-api`.

AWS CLI

Pour créer une API

Commande :

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first
 API'
```

Pour créer une API dupliquée à partir d'une API existante

Commande :

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is
 a copy of my first API' --clone-from 1234123412
```

- Pour plus de détails sur l'API, reportez-vous CreateRestApià la section Référence des AWS CLI commandes.

`create-stage`

L'exemple de code suivant montre comment utiliser`create-stage`.

AWS CLI

Pour créer une étape dans une API qui contiendra un déploiement existant

Commande :

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --
description 'Development stage' --deployment-id a1b2c3
```

Pour créer une étape dans une API qui contiendra un déploiement existant et des variables d'étape personnalisées

Commande :

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'
 --description 'Development stage' --deployment-id a1b2c3 --variables
 key='value',otherKey='otherValue'
```

- Pour plus de détails sur l'API, reportez-vous CreateStageà la section Référence des AWS CLI commandes.

`create-usage-plan-key`

L'exemple de code suivant montre comment utiliser`create-usage-plan-key`.

AWS CLI

Associer une clé d'API existante à un plan d'utilisation

Commande :

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --
key-id 4vq3yryqm5
```

- Pour plus de détails sur l'API, reportez-vous CreateUsagePlanKeyà la section Référence des AWS CLI commandes.

`create-usage-plan`

L'exemple de code suivant montre comment utiliser`create-usage-plan`.

AWS CLI

Pour créer un plan d'utilisation avec des limites d'accélération et de quota qui se réinitialise au début du mois

Commande :

```
aws apigateway create-usage-plan --name "New Usage Plan" --description "A new usage
 plan" --throttle burstLimit=10,rateLimit=5 --quota limit=500,offset=0,period=MONTH
```

- Pour plus de détails sur l'API, reportez-vous CreateUsagePlanà la section Référence des AWS CLI commandes.

`delete-api-key`

L'exemple de code suivant montre comment utiliser`delete-api-key`.

AWS CLI

Pour supprimer une clé d'API

Commande :

```
aws apigateway delete-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

- Pour plus de détails sur l'API, reportez-vous DeleteApiKeyà la section Référence des AWS CLI commandes.

`delete-authorizer`

L'exemple de code suivant montre comment utiliser`delete-authorizer`.

AWS CLI

Pour supprimer un autorisateur personnalisé dans une API

Commande :

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- Pour plus de détails sur l'API, reportez-vous DeleteAuthorizerà la section Référence des AWS CLI commandes.

`delete-base-path-mapping`

L'exemple de code suivant montre comment utiliser`delete-base-path-mapping`.

AWS CLI

Pour supprimer un mappage de chemin de base pour un nom de domaine personnalisé

Commande :

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path
 'dev'
```

- Pour plus de détails sur l'API, reportez-vous DeleteBasePathMappingà la section Référence des AWS CLI commandes.

`delete-client-certificate`

L'exemple de code suivant montre comment utiliser`delete-client-certificate`.

AWS CLI

Pour supprimer un certificat client

Commande :

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous DeleteClientCertificateà la section Référence des AWS CLI commandes.

`delete-deployment`

L'exemple de code suivant montre comment utiliser`delete-deployment`.

AWS CLI

Pour supprimer un déploiement dans une API

Commande :

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous DeleteDeploymentà la section Référence des AWS CLI commandes.

`delete-domain-name-access-association`

L'exemple de code suivant montre comment utiliser`delete-domain-name-access-association`.

AWS CLI

Pour supprimer une association d'accès à un nom de domaine

L'`delete-domain-name-access-association`exemple suivant supprime une association d'accès au nom de domaine entre un nom de domaine personnalisé privé et un point de terminaison VPC.

```
aws apigateway delete-domain-name-access-association \
    --domain-name-access-association-arn arn:aws:apigateway:us-west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-abcd1234efg
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDomainNameAccessAssociationà la section Référence des AWS CLI commandes.

`delete-domain-name`

L'exemple de code suivant montre comment utiliser`delete-domain-name`.

AWS CLI

Pour supprimer un nom de domaine personnalisé

Commande :

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- Pour plus de détails sur l'API, reportez-vous DeleteDomainNameà la section Référence des AWS CLI commandes.

`delete-integration-response`

L'exemple de code suivant montre comment utiliser`delete-integration-response`.

AWS CLI

Pour supprimer une réponse d'intégration pour une ressource, une méthode et un code d'état donnés dans une API

Commande :

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200
```

- Pour plus de détails sur l'API, reportez-vous DeleteIntegrationResponseà la section Référence des AWS CLI commandes.

`delete-integration`

L'exemple de code suivant montre comment utiliser`delete-integration`.

AWS CLI

Pour supprimer une intégration pour une ressource et une méthode données dans une API

Commande :

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- Pour plus de détails sur l'API, reportez-vous DeleteIntegrationà la section Référence des AWS CLI commandes.

`delete-method-response`

L'exemple de code suivant montre comment utiliser`delete-method-response`.

AWS CLI

Pour supprimer une réponse de méthode pour la ressource, la méthode et le code d'état donnés dans une API

Commande :

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3
 --http-method GET --status-code 200
```

- Pour plus de détails sur l'API, reportez-vous DeleteMethodResponseà la section Référence des AWS CLI commandes.

`delete-method`

L'exemple de code suivant montre comment utiliser`delete-method`.

AWS CLI

Pour supprimer une méthode pour la ressource donnée dans une API

Commande :

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET
```

- Pour plus de détails sur l'API, reportez-vous DeleteMethodà la section Référence des AWS CLI commandes.

`delete-model`

L'exemple de code suivant montre comment utiliser`delete-model`.

AWS CLI

Pour supprimer un modèle dans l'API donnée

Commande :

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- Pour plus de détails sur l'API, reportez-vous DeleteModelà la section Référence des AWS CLI commandes.

`delete-resource`

L'exemple de code suivant montre comment utiliser`delete-resource`.

AWS CLI

Pour supprimer une ressource dans une API

Commande :

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous DeleteResourceà la section Référence des AWS CLI commandes.

`delete-rest-api`

L'exemple de code suivant montre comment utiliser`delete-rest-api`.

AWS CLI

Pour supprimer une API

Commande :

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- Pour plus de détails sur l'API, reportez-vous DeleteRestApià la section Référence des AWS CLI commandes.

`delete-stage`

L'exemple de code suivant montre comment utiliser`delete-stage`.

AWS CLI

Pour supprimer une étape dans une API

Commande :

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- Pour plus de détails sur l'API, reportez-vous DeleteStageà la section Référence des AWS CLI commandes.

`delete-usage-plan-key`

L'exemple de code suivant montre comment utiliser`delete-usage-plan-key`.

AWS CLI

Pour supprimer une clé d'API d'un plan d'utilisation

Commande :

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- Pour plus de détails sur l'API, reportez-vous DeleteUsagePlanKeyà la section Référence des AWS CLI commandes.

`delete-usage-plan`

L'exemple de code suivant montre comment utiliser`delete-usage-plan`.

AWS CLI

Pour supprimer un plan d'utilisation

Commande :

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous DeleteUsagePlanà la section Référence des AWS CLI commandes.

`flush-stage-authorizers-cache`

L'exemple de code suivant montre comment utiliser`flush-stage-authorizers-cache`.

AWS CLI

Pour vider toutes les entrées du cache de l'autorisateur sur une scène

Commande :

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-
name dev
```

- Pour plus de détails sur l'API, reportez-vous FlushStageAuthorizersCacheà la section Référence des AWS CLI commandes.

`flush-stage-cache`

L'exemple de code suivant montre comment utiliser`flush-stage-cache`.

AWS CLI

Pour vider le cache du stage d'une API

L'`flush-stage-cache`exemple suivant vide le cache d'une étape.

```
aws apigateway flush-stage-cache \
    --rest-api-id 1234123412 \
    --stage-name dev
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Purger le cache du stage d'API dans API Gateway dans le manuel du développeur Amazon API Gateway.

- Pour plus de détails sur l'API, reportez-vous FlushStageCacheà la section Référence des AWS CLI commandes.

`generate-client-certificate`

L'exemple de code suivant montre comment utiliser`generate-client-certificate`.

AWS CLI

Pour créer un certificat SSL côté client

Commande :

```
aws apigateway generate-client-certificate --description 'My First Client
 Certificate'
```

- Pour plus de détails sur l'API, reportez-vous GenerateClientCertificateà la section Référence des AWS CLI commandes.

`get-account`

L'exemple de code suivant montre comment utiliser`get-account`.

AWS CLI

Pour obtenir les paramètres du compte API Gateway

Commande :

```
aws apigateway get-account
```

Sortie :

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogsRole",
    "throttleSettings": {
        "rateLimit": 500.0,
        "burstLimit": 1000
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetAccountà la section Référence des AWS CLI commandes.

`get-api-key`

L'exemple de code suivant montre comment utiliser`get-api-key`.

AWS CLI

Pour obtenir les informations relatives à une clé d'API spécifique

Commande :

```
aws apigateway get-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

Sortie :

```
{
    "description": "My first key",
    "enabled": true,
    "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
    ],
    "lastUpdatedDate": 1456184515,
    "createdDate": 1456184452,
    "id": "8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk",
    "name": "My key"
}
```

- Pour plus de détails sur l'API, reportez-vous GetApiKeyà la section Référence des AWS CLI commandes.

`get-api-keys`

L'exemple de code suivant montre comment utiliser`get-api-keys`.

AWS CLI

Pour obtenir la liste des clés d'API

Commande :

```
aws apigateway get-api-keys
```

Sortie :

```
{
    "items": [
        {
            "description": "My first key",
            "enabled": true,
            "stageKeys": [
                "a1b2c3d4e5/dev",
                "e5d4c3b2a1/dev"
            ],
            "lastUpdatedDate": 1456184515,
```

```
            "createdDate": 1456184452,
            "id": "8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk",
            "name": "My key"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetApiKeysà la section Référence des AWS CLI commandes.

`get-authorizer`

L'exemple de code suivant montre comment utiliser`get-authorizer`.

AWS CLI

Pour obtenir les paramètres de l'autorisateur par API Gateway

Commande :

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```

Sortie :

```
{
    "authorizerResultTtlInSeconds": 300,
    "name": "MyAuthorizer",
    "type": "TOKEN",
    "identitySource": "method.request.header.Authorization",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:authorizer_function/invocations",
    "id": "gfi4n3"
}
```

- Pour plus de détails sur l'API, reportez-vous GetAuthorizerà la section Référence des AWS CLI commandes.

`get-authorizers`

L'exemple de code suivant montre comment utiliser`get-authorizers`.

AWS CLI

Pour obtenir la liste des autorisateurs pour une API REST

Commande :

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

Sortie :

```
{
    "items": [
        {
            "name": "MyAuthorizer",
            "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123412341234:function:My_Authorizer_Function/
invocations",
            "authorizerResultTtlInSeconds": 300,
            "identitySource": "method.request.header.Authorization",
            "type": "TOKEN",
            "id": "gfi4n3"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetAuthorizersà la section Référence des AWS CLI commandes.

`get-base-path-mapping`

L'exemple de code suivant montre comment utiliser`get-base-path-mapping`.

AWS CLI

Pour obtenir le mappage du chemin de base pour un nom de domaine personnalisé

Commande :

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-
path v1
```

Sortie :

```
{
    "basePath": "v1",
    "restApiId": "1234w4321e",
    "stage": "api"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBasePathMappingà la section Référence des AWS CLI commandes.

`get-base-path-mappings`

L'exemple de code suivant montre comment utiliser`get-base-path-mappings`.

AWS CLI

Pour obtenir les mappages de chemins de base pour un nom de domaine personnalisé

Commande :

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

Sortie :

```
{
    "items": [
        {
            "basePath": "(none)",
            "restApiId": "1234w4321e",
            "stage": "dev"
        },
        {
            "basePath": "v1",
            "restApiId": "1234w4321e",
            "stage": "api"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBasePathMappingsà la section Référence des AWS CLI commandes.

`get-client-certificate`

L'exemple de code suivant montre comment utiliser`get-client-certificate`.

AWS CLI

Pour obtenir un certificat client

Commande :

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous GetClientCertificateà la section Référence des AWS CLI commandes.

`get-client-certificates`

L'exemple de code suivant montre comment utiliser`get-client-certificates`.

AWS CLI

Pour obtenir la liste des certificats clients

Commande :

```
aws apigateway get-client-certificates
```

Sortie :

```
{
    "items": [
        {
            "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
 content> -----END CERTIFICATE-----",
            "clientCertificateId": "a1b2c3",
            "expirationDate": 1483556561,
            "description": "My Client Certificate",
            "createdDate": 1452020561
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetClientCertificatesà la section Référence des AWS CLI commandes.

`get-deployment`

L'exemple de code suivant montre comment utiliser`get-deployment`.

AWS CLI

Pour obtenir des informations sur un déploiement

Commande :

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

Sortie :

```
{
    "description": "myDeployment",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentà la section Référence des AWS CLI commandes.

`get-deployments`

L'exemple de code suivant montre comment utiliser`get-deployments`.

AWS CLI

Pour obtenir la liste des déploiements d'une API REST

Commande :

```
aws apigateway get-deployments --rest-api-id 1234123412
```

Sortie :

```
{
    "items": [
        {
            "createdDate": 1453797217,
            "id": "0a2b4c",
            "description": "Deployed my API for the first time"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentsà la section Référence des AWS CLI commandes.

`get-domain-name-access-associations`

L'exemple de code suivant montre comment utiliser`get-domain-name-access-associations`.

AWS CLI

Exemple 1 : pour répertorier toutes les associations d'accès aux noms de domaine

L'`get-domain-name-access-associations`exemple suivant répertorie toutes les associations d'accès aux noms de domaine.

```
aws apigateway get-domain-name-access-associations
```

Sortie :

```
{
    "items": [
        {
        "domainNameAccessAssociationArn": "arn:aws:apigateway:us-
west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/
vpcesource/vpce-abcd1234efg
        "accessAssociationSource": "vpce-abcd1234efg",
        "accessAssociationSourceType": "VPCE",
        "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
        }
    ]
}
```

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Exemple 2 : pour répertorier toutes les associations d'accès aux noms de domaine détenues par ce AWS compte

L'`get-domain-name-access-associations`exemple suivant répertorie toutes les associations d'accès aux noms de domaine détenues par le AWS compte courant.

```
aws apigateway get-domain-name-access-associations \
    --resource-owner SELF
```

Sortie :

```
{
    "items": [
        {
        "domainNameAccessAssociationArn": "arn:aws:apigateway:us-
west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/
vpcesource/vpce-abcd1234efg
        "accessAssociationSource": "vpce-abcd1234efg",
        "accessAssociationSourceType": "VPCE",
        "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
        }
    ]
}
```

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDomainNameAccessAssociationsà la section Référence des AWS CLI commandes.

`get-domain-name`

L'exemple de code suivant montre comment utiliser`get-domain-name`.

AWS CLI

Exemple 1 : Pour obtenir des informations sur un nom de domaine public personnalisé

L'`get-domain-name`exemple suivant permet d'obtenir des informations sur un nom de domaine public personnalisé.

```
aws apigateway get-domain-name \
    --domain-name api.domain.tld
```

Sortie :

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
    "certificateName": "uploadedCertificate",
    "certificateUploadDate": 1462565487
}
```

Pour plus d'informations, consultez la section Nom de domaine personnalisé pour le REST public APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Exemple 2 : Pour obtenir des informations sur un nom de domaine personnalisé privé

L'`get-domain-name`exemple suivant permet d'obtenir des informations sur un nom de domaine personnalisé privé.

```
aws apigateway get-domain-name \
    --domain-name api.private.domain.tld \
    --domain-name-id abcd1234
```

Sortie :

```
{
    "domainName": "my.private.domain.tld",
    "domainNameId": "abcd1234",
    "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-09-10T10:31:20-07:00",
    "endpointConfiguration": {
        "types": [
            "PRIVATE"
        ]
```

```
    },
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":\"*\",\"Action\":\"execute-api:Invoke\",\"Resource\":
\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/my.private.domain.tld
+abcd1234\"},{\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"execute-
api:Invoke\",\"Resource\":\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234\",\"Condition\":{\"StringNotEquals\":{\"aws:SourceVpc
\":\"vpc-1a2b3c4d\"}}}]}"
}
```

Pour plus d'informations, consultez la section Nom de domaine personnalisé pour le REST public APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDomainNameà la section Référence des AWS CLI commandes.

`get-domain-names`

L'exemple de code suivant montre comment utiliser`get-domain-names`.

AWS CLI

Exemple 1 : pour obtenir une liste de noms de domaine personnalisés

La `get-domain-names` commande suivante permet d'obtenir une liste de noms de domaine.

```
aws apigateway get-domain-names
```

Sortie :

```
{
    "items": [
        {
            "distributionDomainName": "d9511k3l09bkd.cloudfront.net",
            "certificateUploadDate": 1452812505,
            "certificateName": "my_custom_domain-certificate",
            "domainName": "subdomain.domain.tld"
        }
    ]
}
```

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Exemple 2 : pour obtenir une liste des noms de domaine personnalisés détenus par ce AWS compte

La `get-domain-names` commande suivante permet d'obtenir la liste des noms de domaine détenus par ce AWS compte.

```
aws apigateway get-domain-names \
    --resource-owner SELF
```

Sortie :

```
{
    "items": [
        {
            "domainName": "my.domain.tld",
            "domainNameArn": "arn:aws:apigateway:us-east-1::/domainnames/
my.private.domain.tld",
            "certificateUploadDate": "2024-08-15T17:02:55-07:00",
            "regionalDomainName": "d-abcd1234.execute-api.us-east-1.amazonaws.com",
            "regionalHostedZoneId": "Z1UJRXOUMOOFQ8",
            "regionalCertificateArn": "arn:aws:acm:us-
east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3",
            "endpointConfiguration": {
                "types": [
                    "REGIONAL"
                ]
            },
            "domainNameStatus": "AVAILABLE",
            "securityPolicy": "TLS_1_2"
        },
        {
            "domainName": "my.private.domain.tld",
            "domainNameId": "abcd1234",
            "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/
domainnames/my.private.domain.tld+abcd1234",
            "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/
fb1b9770-a305-495d-aefb-27e5e101ff3",
            "certificateUploadDate": "2024-11-26T11:44:40-08:00",
            "endpointConfiguration": {
```

```
                "types": [
                    "PRIVATE"
                ]
            },
            "domainNameStatus": "AVAILABLE",
            "securityPolicy": "TLS_1_2"
        }
    ]
}
```

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Exemple 3 : pour obtenir une liste de noms de domaine personnalisés appartenant à d'autres AWS comptes avec lesquels vous pouvez créer une association d'accès aux noms de domaine.

La `get-domain-names` commande suivante permet d'obtenir la liste des noms de domaine détenus par d'autres AWS comptes auxquels vous avez accès pour créer une association d'accès aux noms de domaine.

```
aws apigateway get-domain-names \
    --resource-owner OTHER_ACCOUNTS
```

Sortie :

```
{
    "items": [
        {
            "domainName": "my.private.domain.tld",
            "domainNameId": "abcd1234",
            "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/
domainnames/my.private.domain.tld+abcd1234"
        }
    ]
}
```

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDomainNamesà la section Référence des AWS CLI commandes.

`get-export`

L'exemple de code suivant montre comment utiliser`get-export`.

AWS CLI

Pour obtenir le modèle JSON Swagger pour une étape

Commande :

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-
type swagger /path/to/filename.json
```

Pour obtenir le modèle JSON Swagger et les extensions API Gateway pour une étape

Commande :

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-
id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Pour obtenir le modèle JSON Swagger et les extensions Postman pour une étape

Commande :

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5
 --stage-name dev --export-type swagger /path/to/filename.json
```

- Pour plus de détails sur l'API, reportez-vous GetExportà la section Référence des AWS CLI commandes.

`get-integration-response`

L'exemple de code suivant montre comment utiliser`get-integration-response`.

AWS CLI

Pour obtenir la configuration de la réponse d'intégration pour une méthode HTTP définie dans la ressource d'une API REST

Commande :

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-
id y9h6rt --http-method GET --status-code 200
```

Sortie :

```
{
    "statusCode": "200",
    "responseTemplates": {
        "application/json": null
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetIntegrationResponseà la section Référence des AWS CLI commandes.

`get-integration`

L'exemple de code suivant montre comment utiliser`get-integration`.

AWS CLI

Pour obtenir la configuration d'intégration d'une méthode HTTP définie dans la ressource d'une API REST

Commande :

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-
method GET
```

Sortie :

```
{
    "httpMethod": "POST",
    "integrationResponses": {
        "200": {
            "responseTemplates": {
                "application/json": null
            },
            "statusCode": "200"
        }
```

```
    },
    "cacheKeyParameters": [],
    "type": "AWS",
    "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
    "cacheNamespace": "y9h6rt"
}
```

- Pour plus de détails sur l'API, reportez-vous GetIntegrationà la section Référence des AWS CLI commandes.

`get-method-response`

L'exemple de code suivant montre comment utiliser`get-method-response`.

AWS CLI

Pour obtenir la configuration de la ressource de réponse d'une méthode HTTP définie dans la ressource d'une API REST

Commande :

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --
http-method GET --status-code 200
```

Sortie :

```
{
    "responseModels": {
        "application/json": "Empty"
    },
    "statusCode": "200"
}
```

- Pour plus de détails sur l'API, reportez-vous GetMethodResponseà la section Référence des AWS CLI commandes.

`get-method`

L'exemple de code suivant montre comment utiliser`get-method`.

AWS CLI

Pour obtenir la configuration des ressources de méthode pour une méthode HTTP définie dans la ressource d'une API REST

Commande :

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-
method GET
```

Sortie :

```
{
    "apiKeyRequired": false,
    "httpMethod": "GET",
    "methodIntegration": {
        "integrationResponses": {
            "200": {
                "responseTemplates": {
                    "application/json": null
                },
                "statusCode": "200"
            }
        },
        "cacheKeyParameters": [],
        "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
        "httpMethod": "POST",
        "cacheNamespace": "y9h6rt",
        "type": "AWS"
    },
    "requestParameters": {},
    "methodResponses": {
        "200": {
            "responseModels": {
                "application/json": "Empty"
            },
            "statusCode": "200"
        }
    },
    "authorizationType": "NONE"
}
```

- Pour plus de détails sur l'API, reportez-vous GetMethodà la section Référence des AWS CLI commandes.

`get-model-template`

L'exemple de code suivant montre comment utiliser`get-model-template`.

AWS CLI

Pour obtenir le modèle de mappage d'un modèle défini sous une API REST

Commande :

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

Sortie :

```
{
    "value": "#set($inputRoot = $input.path('$'))\n{ }"
}
```

- Pour plus de détails sur l'API, reportez-vous GetModelTemplateà la section Référence des AWS CLI commandes.

`get-model`

L'exemple de code suivant montre comment utiliser`get-model`.

AWS CLI

Pour obtenir la configuration d'un modèle défini sous une API REST

Commande :

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

Sortie :

```
{
    "contentType": "application/json",
    "description": "This is a default empty schema model",
```

```
    "name": "Empty",
    "id": "etd5w5",
    "schema": "{\n  \"$schema\": \"http://json-schema.org/draft-04/schema#\",\n
 \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}"
}
```

- Pour plus de détails sur l'API, reportez-vous GetModelà la section Référence des AWS CLI commandes.

`get-models`

L'exemple de code suivant montre comment utiliser`get-models`.

AWS CLI

Pour obtenir une liste de modèles pour une API REST

Commande :

```
aws apigateway get-models --rest-api-id 1234123412
```

Sortie :

```
{
    "items": [
        {
            "description": "This is a default error schema model",
            "schema": "{\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#
\",\n  \"title\" : \"Error Schema\",\n  \"type\" : \"object\",\n  \"properties\" :
 {\n    \"message\" : { \"type\" : \"string\" }\n  }\n}",
            "contentType": "application/json",
            "id": "7tpbze",
            "name": "Error"
        },
        {
            "description": "This is a default empty schema model",
            "schema": "{\n  \"$schema\": \"http://json-schema.org/draft-04/schema#
\",\n  \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}",
            "contentType": "application/json",
            "id": "etd5w5",
            "name": "Empty"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetModelsà la section Référence des AWS CLI commandes.

`get-resource`

L'exemple de code suivant montre comment utiliser`get-resource`.

AWS CLI

Pour obtenir des informations sur une ressource

Commande :

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

Sortie :

```
{
    "path": "/path",
    "pathPart": "path",
    "id": "zwo0y3",
    "parentId": "uyokt6ij2g"
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourceà la section Référence des AWS CLI commandes.

`get-resources`

L'exemple de code suivant montre comment utiliser`get-resources`.

AWS CLI

Pour obtenir une liste de ressources pour une API REST

Commande :

```
aws apigateway get-resources --rest-api-id 1234123412
```

Sortie :

```
{
    "items": [
        {
            "path": "/resource/subresource",
            "resourceMethods": {
                "POST": {}
            },
            "id": "024ace",
            "pathPart": "subresource",
            "parentId": "ai5b02"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourcesà la section Référence des AWS CLI commandes.

`get-rest-api`

L'exemple de code suivant montre comment utiliser`get-rest-api`.

AWS CLI

Pour obtenir des informations sur une API

Commande :

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

Sortie :

```
{
    "name": "myAPI",
    "id": "o1y243m4f5",
    "createdDate": 1453416433
}
```

- Pour plus de détails sur l'API, reportez-vous GetRestApià la section Référence des AWS CLI commandes.

`get-rest-apis`

L'exemple de code suivant montre comment utiliser`get-rest-apis`.

AWS CLI

Pour obtenir une liste de REST APIs

Commande :

```
aws apigateway get-rest-apis
```

Sortie :

```
{
    "items": [
        {
            "createdDate": 1438884790,
            "id": "12s44z21rb",
            "name": "My First API"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRestApisà la section Référence des AWS CLI commandes.

`get-sdk`

L'exemple de code suivant montre comment utiliser`get-sdk`.

AWS CLI

Pour obtenir le SDK Android pour une étape d'API REST

Commande :

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-type android
 --parameters
 groupId='com.mycompany',invokerPackage='com.mycompany.clientsdk',artifactId='Mycompany-
client',artifactVersion='1.0.0' /path/to/android_sdk.zip
```

Sortie :

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}
```

Pour obtenir le SDK IOS pour une étape d'API REST

Commande :

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type objectivec --parameters classPrefix='myprefix' /path/to/iOS_sdk.zip
```

Sortie :

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip
\""
}
```

Pour obtenir le SDK Javascript pour une étape d'API REST

Commande :

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type javascript /path/to/javascript_sdk.zip
```

Sortie :

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip
\""
}
```

- Pour plus de détails sur l'API, reportez-vous GetSdkà la section Référence des AWS CLI commandes.

`get-stage`

L'exemple de code suivant montre comment utiliser`get-stage`.

AWS CLI

Pour obtenir des informations sur le stage d'une API

Commande :

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

Sortie :

```
{
    "stageName": "dev",
    "cacheClusterSize": "0.5",
    "cacheClusterEnabled": false,
    "cacheClusterStatus": "NOT_AVAILABLE",
    "deploymentId": "rbh1fj",
    "lastUpdatedDate": 1466802961,
    "createdDate": 1460682074,
    "methodSettings": {
        "*/*": {
            "cacheTtlInSeconds": 300,
            "loggingLevel": "INFO",
            "dataTraceEnabled": false,
            "metricsEnabled": true,
            "unauthorizedCacheControlHeaderStrategy":
 "SUCCEED_WITH_RESPONSE_HEADER",
            "throttlingRateLimit": 500.0,
            "cacheDataEncrypted": false,
            "cachingEnabled": false,
            "throttlingBurstLimit": 1000,
            "requireAuthorizationForCacheControl": true
        },
        "~1resource/GET": {
            "cacheTtlInSeconds": 300,
            "loggingLevel": "INFO",
            "dataTraceEnabled": false,
            "metricsEnabled": true,
            "unauthorizedCacheControlHeaderStrategy":
 "SUCCEED_WITH_RESPONSE_HEADER",
```

```
            "throttlingRateLimit": 500.0,
            "cacheDataEncrypted": false,
            "cachingEnabled": false,
            "throttlingBurstLimit": 1000,
            "requireAuthorizationForCacheControl": true
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetStageà la section Référence des AWS CLI commandes.

`get-stages`

L'exemple de code suivant montre comment utiliser`get-stages`.

AWS CLI

Pour obtenir la liste des étapes d'une API REST

Commande :

```
aws apigateway get-stages --rest-api-id 1234123412
```

Sortie :

```
{
    "item": [
        {
            "stageName": "dev",
            "cacheClusterSize": "0.5",
            "cacheClusterEnabled": true,
            "cacheClusterStatus": "AVAILABLE",
            "deploymentId": "123h64",
            "lastUpdatedDate": 1456185138,
            "createdDate": 1453589092,
            "methodSettings": {
                "~1resource~1subresource/POST": {
                    "cacheTtlInSeconds": 300,
                    "loggingLevel": "INFO",
                    "dataTraceEnabled": true,
                    "metricsEnabled": true,
```

```
                    "throttlingRateLimit": 500.0,
                    "cacheDataEncrypted": false,
                    "cachingEnabled": false,
                    "throttlingBurstLimit": 1000
                }
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetStagesà la section Référence des AWS CLI commandes.

`get-usage-plan-key`

L'exemple de code suivant montre comment utiliser`get-usage-plan-key`.

AWS CLI

Pour obtenir les détails d'une clé d'API associée à un plan d'utilisation

Commande :

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- Pour plus de détails sur l'API, reportez-vous GetUsagePlanKeyà la section Référence des AWS CLI commandes.

`get-usage-plan-keys`

L'exemple de code suivant montre comment utiliser`get-usage-plan-keys`.

AWS CLI

Pour obtenir la liste des clés d'API associées à un plan d'utilisation

Commande :

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous GetUsagePlanKeysà la section Référence des AWS CLI commandes.

`get-usage-plan`

L'exemple de code suivant montre comment utiliser`get-usage-plan`.

AWS CLI

Pour obtenir les détails d'un plan d'utilisation

Commande :

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- Pour plus de détails sur l'API, reportez-vous GetUsagePlanà la section Référence des AWS CLI commandes.

`get-usage-plans`

L'exemple de code suivant montre comment utiliser`get-usage-plans`.

AWS CLI

Pour obtenir les détails de tous les plans d'utilisation

Commande :

```
aws apigateway get-usage-plans
```

- Pour plus de détails sur l'API, reportez-vous GetUsagePlansà la section Référence des AWS CLI commandes.

`get-usage`

L'exemple de code suivant montre comment utiliser`get-usage`.

AWS CLI

Pour obtenir les détails d'utilisation d'un plan d'utilisation

Commande :

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-
date "2016-08-17"
```

- Pour plus de détails sur l'API, reportez-vous GetUsageà la section Référence des AWS CLI commandes.

`import-rest-api`

L'exemple de code suivant montre comment utiliser`import-rest-api`.

AWS CLI

Pour importer un modèle Swagger et créer une API

Commande :

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- Pour plus de détails sur l'API, reportez-vous ImportRestApià la section Référence des AWS CLI commandes.

`put-integration-response`

L'exemple de code suivant montre comment utiliser`put-integration-response`.

AWS CLI

Pour créer une réponse d'intégration comme réponse par défaut avec un modèle de mappage défini

Commande :

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200 --selection-pattern "" --response-
templates '{"application/json": "{\"json\": \"template\"}"}'
```

Pour créer une réponse d'intégration avec une expression régulière de 400 et une valeur d'en-tête définie de manière statique

Commande :

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 400 --selection-pattern 400 --response-
parameters '{"method.response.header.custom-header": "'"'"'custom-value'"'"'"}'
```

- Pour plus de détails sur l'API, reportez-vous PutIntegrationResponseà la section Référence des AWS CLI commandes.

`put-integration`

L'exemple de code suivant montre comment utiliser`put-integration`.

AWS CLI

Pour créer une demande d'intégration MOCK

Commande :

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET --type MOCK --request-templates '{ "application/json": "{\"statusCode\":
 200}" }'
```

Pour créer une demande d'intégration HTTP

Commande :

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET --type HTTP --integration-http-method GET --uri 'https://domain.tld/path'
```

Pour créer une demande d' AWS intégration avec un point de terminaison de fonction Lambda

Commande :

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --type AWS --integration-http-method POST --uri
 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-
west-2:123412341234:function:function_name/invocations'
```

- Pour plus de détails sur l'API, reportez-vous PutIntegrationà la section Référence des AWS CLI commandes.

`put-method-response`

L'exemple de code suivant montre comment utiliser`put-method-response`.

AWS CLI

Pour créer une réponse de méthode sous le code d'état spécifié avec un en-tête de réponse de méthode personnalisé

Commande :

```
aws apigateway put-method-response --rest-api-id 1234123412 --
resource-id a1b2c3 --http-method GET --status-code 400 --response-
parameters "method.response.header.custom-header=false"
```

- Pour plus de détails sur l'API, reportez-vous PutMethodResponseà la section Référence des AWS CLI commandes.

`put-method`

L'exemple de code suivant montre comment utiliser`put-method`.

AWS CLI

Pour créer une méthode pour une ressource dans une API sans autorisation, sans clé d'API et avec un en-tête de demande de méthode personnalisé

Commande :

```
aws apigateway put-method --rest-api-id 1234123412 --resource-id a1b2c3 --
http-method PUT --authorization-type "NONE" --no-api-key-required --request-
parameters "method.request.header.custom-header=false"
```

- Pour plus de détails sur l'API, reportez-vous PutMethodà la section Référence des AWS CLI commandes.

`put-rest-api`

L'exemple de code suivant montre comment utiliser`put-rest-api`.

AWS CLI

Pour remplacer une API existante à l'aide d'un modèle Swagger

Commande :

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body
 'fileb:///path/to/API_Swagger_template.json'
```

Pour fusionner un modèle Swagger dans une API existante

Commande :

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///
path/to/API_Swagger_template.json'
```

- Pour plus de détails sur l'API, reportez-vous PutRestApià la section Référence des AWS CLI commandes.

`reject-domain-name-access-association`

L'exemple de code suivant montre comment utiliser`reject-domain-name-access-association`.

AWS CLI

Pour rejeter une association d'accès à un nom de domaine

L'`reject-domain-name-access-association`exemple suivant rejette une association d'accès au nom de domaine entre un nom de domaine personnalisé privé et un point de terminaison VPC.

```
aws apigateway reject-domain-name-access-association \
    --domain-name-access-association-arn arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg \
    --domain-name-arn arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Noms de domaine personnalisés pour le privé APIs dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RejectDomainNameAccessAssociationà la section Référence des AWS CLI commandes.

`test-invoke-authorizer`

L'exemple de code suivant montre comment utiliser`test-invoke-authorizer`.

AWS CLI

Pour tester, appelez une demande à un autorisateur personnalisé, y compris l'en-tête et la valeur requis.

Commande :

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-
id 5yid1t --headers Authorization='Value'
```

- Pour plus de détails sur l'API, reportez-vous TestInvokeAuthorizerà la section Référence des AWS CLI commandes.

`test-invoke-method`

L'exemple de code suivant montre comment utiliser`test-invoke-method`.

AWS CLI

Pour tester, invoquez la ressource racine dans une API en effectuant une requête GET

Commande :

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id avl5sg8fw8
 --http-method GET --path-with-query-string '/'
```

Pour tester, invoquez une sous-ressource dans une API en effectuant une requête GET avec une valeur de paramètre de chemin spécifiée.

Commande :

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- Pour plus de détails sur l'API, reportez-vous TestInvokeMethodà la section Référence des AWS CLI commandes.

`update-account`

L'exemple de code suivant montre comment utiliser`update-account`.

AWS CLI

Pour modifier l'ARN du rôle IAM pour la connexion aux journaux CloudWatch

Commande :

```
aws apigateway update-account --patch-operations op='replace',path='/
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

Sortie :

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogs",
    "throttleSettings": {
        "rateLimit": 1000.0,
        "burstLimit": 2000
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateAccountà la section Référence des AWS CLI commandes.

`update-api-key`

L'exemple de code suivant montre comment utiliser`update-api-key`.

AWS CLI

Pour modifier le nom d'une clé d'API

Commande :

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/name',value='newName'
```

Sortie :

```
{
    "description": "currentDescription",
    "enabled": true,
    "stageKeys": [
        "41t2j324r5/dev"
    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
}
```

Pour désactiver la clé d'API

Commande :

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/enabled',value='false'
```

Sortie :

```
{
    "description": "currentDescription",
    "enabled": false,
    "stageKeys": [
        "41t2j324r5/dev"
    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateApiKeyà la section Référence des AWS CLI commandes.

`update-authorizer`

L'exemple de code suivant montre comment utiliser`update-authorizer`.

AWS CLI

Pour modifier le nom de l'autorisateur personnalisé

Commande :

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/name',value='testAuthorizer'
```

Sortie :

```
{
    "authType": "custom",
    "name": "testAuthorizer",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthorizer/invocations",
    "authorizerResultTtlInSeconds": 300,
    "identitySource": "method.request.header.Authorization",
    "type": "TOKEN",
    "id": "gfi4n3"
}
```

Pour modifier la fonction Lambda invoquée par l'autorisateur personnalisé

Commande :

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/authorizerUri',value='arn:aws:apigateway:us-
west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-
west-2:123412341234:function:newAuthorizer/invocations'
```

Sortie :

```
{
    "authType": "custom",
    "name": "testAuthorizer",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:newAuthorizer/invocations",
```

```
    "authorizerResultTtlInSeconds": 300,
    "identitySource": "method.request.header.Authorization",
    "type": "TOKEN",
    "id": "gfi4n3"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateAuthorizerà la section Référence des AWS CLI commandes.

`update-base-path-mapping`

L'exemple de code suivant montre comment utiliser`update-base-path-mapping`.

AWS CLI

Pour modifier le chemin de base d'un nom de domaine personnalisé

Commande :

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

Sortie :

```
{
    "basePath": "v1",
    "restApiId": "1234123412",
    "stage": "api"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateBasePathMappingà la section Référence des AWS CLI commandes.

`update-client-certificate`

L'exemple de code suivant montre comment utiliser`update-client-certificate`.

AWS CLI

Pour mettre à jour la description d'un certificat client

Commande :

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-operations op='replace',path='/description',value='My new description'
```

- Pour plus de détails sur l'API, reportez-vous UpdateClientCertificateà la section Référence des AWS CLI commandes.

`update-deployment`

L'exemple de code suivant montre comment utiliser`update-deployment`.

AWS CLI

Pour modifier la description d'un déploiement

Commande :

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --patch-operations op='replace',path='/description',value='newDescription'
```

Sortie :

```
{
    "description": "newDescription",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateDeploymentà la section Référence des AWS CLI commandes.

`update-domain-name`

L'exemple de code suivant montre comment utiliser`update-domain-name`.

AWS CLI

Pour modifier le nom du certificat pour un nom de domaine personnalisé

L'`update-domain-name`exemple suivant modifie le nom du certificat d'un domaine personnalisé.

```
aws apigateway update-domain-name \
    --domain-name api.domain.tld \
    --patch-operations op='replace',path='/certificateArn',value='arn:aws:acm:us-
west-2:111122223333:certificate/CERTEXAMPLE123EXAMPLE'
```

Sortie :

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d123456789012.cloudfront.net",
    "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/
CERTEXAMPLE123EXAMPLE",
    "certificateUploadDate": 1462565487
}
```

Pour plus d'informations, consultez Configurer un nom de domaine personnalisé pour une API dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainNameà la section Référence des AWS CLI commandes.

`update-integration-response`

L'exemple de code suivant montre comment utiliser`update-integration-response`.

AWS CLI

Pour modifier un en-tête de réponse d'intégration afin d'obtenir un mappage statique de « * »

Commande :

```
aws apigateway update-integration-response --rest-api-id 1234123412 --
resource-id 3gapai --http-method GET --status-code 200 --patch-operations
 op='replace',path='/responseParameters/method.response.header.Access-Control-Allow-
Origin',value='"'"'*'"'"'
```

Sortie :

```
{
```

```
    "statusCode": "200",
    "responseParameters": {
        "method.response.header.Access-Control-Allow-Origin": "'*'"
    }
}
```

Pour supprimer un en-tête de réponse d'intégration

Commande :

```
aws apigateway update-integration-response --rest-api-id 1234123412 --resource-
id 3gapai --http-method GET --status-code 200 --patch-operations op='remove',path='/
responseParameters/method.response.header.Access-Control-Allow-Origin'
```

- Pour plus de détails sur l'API, reportez-vous UpdateIntegrationResponseà la section Référence des AWS CLI commandes.

`update-integration`

L'exemple de code suivant montre comment utiliser`update-integration`.

AWS CLI

Pour ajouter le modèle de mappage « Content-Type : application/json » configuré avec Input Passthrough

Commande :

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='add',path='/requestTemplates/application~1json'"
```

Pour mettre à jour (remplacer) le modèle de mappage « Content-Type : application/json » configuré par un modèle personnalisé

Commande :

```
aws apigateway update-integration \
```

```
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='replace',path='/requestTemplates/
application~1json',value='{"example": "json"}'"
```

Pour mettre à jour (remplacer) un modèle personnalisé associé à « Content-Type : application/json » avec Input Passthrough

Commande :

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

Pour supprimer le modèle de mappage « Content-Type : application/json »

Commande :

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- Pour plus de détails sur l'API, reportez-vous UpdateIntegrationà la section Référence des AWS CLI commandes.

`update-method-response`

L'exemple de code suivant montre comment utiliser`update-method-response`.

AWS CLI

Pour créer un nouvel en-tête de réponse de méthode pour la réponse 200 d'une méthode et le définir comme non obligatoire (par défaut)

Commande :

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200 --patch-operations op="add",path="/
responseParameters/method.response.header.custom-header",value="false"
```

Pour supprimer un modèle de réponse pour la réponse 200 dans une méthode

Commande :

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-id a1b2c3
 --http-method GET --status-code 200 --patch-operations op="remove",path="/
responseModels/application~1json"
```

- Pour plus de détails sur l'API, reportez-vous UpdateMethodResponseà la section Référence des AWS CLI commandes.

`update-method`

L'exemple de code suivant montre comment utiliser`update-method`.

AWS CLI

Exemple 1 : pour modifier une méthode afin d'exiger une clé d'API

L'`update-method`exemple suivant modifie la méthode pour exiger une clé d'API.

```
aws apigateway update-method \
    --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

Sortie :

```
{
    "httpMethod": "GET",
    "authorizationType": "NONE",
    "apiKeyRequired": true,
    "methodResponses": {
        "200": {
            "statusCode": "200",
            "responseModels": {}
```

```
        }
    },
    "methodIntegration": {
        "type": "AWS",
        "httpMethod": "POST",
        "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
        "passthroughBehavior": "WHEN_NO_MATCH",
        "contentHandling": "CONVERT_TO_TEXT",
        "timeoutInMillis": 29000,
        "cacheNamespace": "h7i8j9",
        "cacheKeyParameters": [],
        "integrationResponses": {
            "200": {
                "statusCode": "200",
                "responseTemplates": {}
            }
        }
    }
}
```

Exemple 2 : pour modifier une méthode afin d'exiger une autorisation IAM

L'`update-method`exemple suivant modifie la méthode pour exiger une autorisation IAM.

```
aws apigateway update-method \
    --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"
```

Sortie :

```
 {
    "httpMethod": "GET",
    "authorizationType": "AWS_IAM",
    "apiKeyRequired": false,
    "methodResponses": {
        "200": {
            "statusCode": "200",
            "responseModels": {}
        }
    },
```

```
    "methodIntegration": {
        "type": "AWS",
        "httpMethod": "POST",
        "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
        "passthroughBehavior": "WHEN_NO_MATCH",
        "contentHandling": "CONVERT_TO_TEXT",
        "timeoutInMillis": 29000,
        "cacheNamespace": "h7i8j9",
        "cacheKeyParameters": [],
        "integrationResponses": {
            "200": {
                "statusCode": "200",
                "responseTemplates": {}
            }
        }
    }
}
```

Exemple 3 : Pour modifier une méthode afin d'exiger une autorisation Lambda

L'`update-method`exemple suivant modifie la méthode pour obtenir l'autorisation Lambda requise.

```
aws apigateway update-method --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/authorizationType",value="CUSTOM"
 op="replace",path="/authorizerId",value="e4f5g6"
```

Sortie :

```
 {
    "httpMethod": "GET",
    "authorizationType": "CUSTOM",
    "authorizerId" : "e4f5g6",
    "apiKeyRequired": false,
    "methodResponses": {
        "200": {
            "statusCode": "200",
            "responseModels": {}
        }
    },
```

```
    "methodIntegration": {
        "type": "AWS",
        "httpMethod": "POST",
        "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
        "passthroughBehavior": "WHEN_NO_MATCH",
        "contentHandling": "CONVERT_TO_TEXT",
        "timeoutInMillis": 29000,
        "cacheNamespace": "h7i8j9",
        "cacheKeyParameters": [],
        "integrationResponses": {
            "200": {
                "statusCode": "200",
                "responseTemplates": {}
            }
        }
    }
}
```

Pour plus d'informations, consultez Créer, configurer et tester des plans d'utilisation à l'aide de l'API Gateway CLI et de l'API REST et Contrôler et gérer l'accès à une API REST dans API Gateway dans le manuel du développeur Amazon API Gateway.

- Pour plus de détails sur l'API, reportez-vous UpdateMethodà la section Référence des AWS CLI commandes.

`update-model`

L'exemple de code suivant montre comment utiliser`update-model`.

AWS CLI

Pour modifier la description d'un modèle dans une API

Commande :

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-
operations op=replace,path=/description,value='New Description'
```

Pour modifier le schéma d'un modèle dans une API

Commande :

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-
operations op=replace,path=/schema,value='"{ \"$schema\": \"http://json-schema.org/
draft-04/schema#\", \"title\" : \"Empty Schema\", \"type\" : \"object\" }"'
```

- Pour plus de détails sur l'API, reportez-vous UpdateModelà la section Référence des AWS CLI commandes.

`update-resource`

L'exemple de code suivant montre comment utiliser`update-resource`.

AWS CLI

Pour déplacer une ressource et la placer sous une autre ressource parent dans une API

Commande :

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --
patch-operations op=replace,path=/parentId,value='3c2b1a'
```

Sortie :

```
{
    "path": "/resource",
    "pathPart": "resource",
    "id": "1a2b3c",
    "parentId": "3c2b1a"
}
```

Pour renommer une ressource (PathPart) dans une API

Commande :

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --
patch-operations op=replace,path=/pathPart,value=newresourcename
```

Sortie :

```
{
    "path": "/newresourcename",
```

```
    "pathPart": "newresourcename",
    "id": "1a2b3c",
    "parentId": "3c2b1a"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateResourceà la section Référence des AWS CLI commandes.

`update-rest-api`

L'exemple de code suivant montre comment utiliser`update-rest-api`.

AWS CLI

Pour modifier le nom d'une API

Commande :

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/name,value='New Name'
```

Pour modifier la description d'une API

Commande :

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/description,value='New Description'
```

- Pour plus de détails sur l'API, reportez-vous UpdateRestApià la section Référence des AWS CLI commandes.

`update-stage`

L'exemple de code suivant montre comment utiliser`update-stage`.

AWS CLI

Exemple 1 : pour remplacer les paramètres d'étape d'une ressource et d'une méthode

L'`update-stage`exemple suivant remplace les paramètres de l'étape et désactive la request/response journalisation complète pour une ressource et une méthode spécifiques.

```
aws apigateway update-stage \
    --rest-api-id 1234123412 \
    --stage-name 'dev' \
    --patch-operations op=replace,path=/~1resourceName/GET/logging/
dataTrace,value=false
```

Sortie :

```
{
    "deploymentId": "5ubd17",
    "stageName": "dev",
    "cacheClusterEnabled": false,
    "cacheClusterStatus": "NOT_AVAILABLE",
    "methodSettings": {
        "~1resourceName/GET": {
            "metricsEnabled": false,
            "dataTraceEnabled": false,
            "throttlingBurstLimit": 5000,
            "throttlingRateLimit": 10000.0,
            "cachingEnabled": false,
            "cacheTtlInSeconds": 300,
            "cacheDataEncrypted": false,
            "requireAuthorizationForCacheControl": true,
            "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"
        }
    },
    "tracingEnabled": false,
    "createdDate": "2022-07-18T10:11:18-07:00",
    "lastUpdatedDate": "2022-07-18T10:19:04-07:00"
}
```

Pour plus d'informations, consultez la section Configuration d'un stage pour une API REST dans le manuel Amazon API Gateway Developer Guide.

Exemple 2 : pour mettre à jour les paramètres d'étape pour toutes les ressources et méthodes d'une étape d'API

L'`update-stage`exemple suivant active la request/response journalisation complète pour toutes les ressources et méthodes d'une étape d'API.

```
aws apigateway update-stage \
    --rest-api-id 1234123412 \
```

```
    --stage-name 'dev' \
    --patch-operations 'op=replace,path=/*/*/logging/dataTrace,value=true'
```

Sortie :

```
{
    "deploymentId": "5ubd17",
    "stageName": "dev",
    "cacheClusterEnabled": false,
    "cacheClusterStatus": "NOT_AVAILABLE",
    "methodSettings": {
        "*/*": {
            "metricsEnabled": false,
            "dataTraceEnabled": true,
            "throttlingBurstLimit": 5000,
            "throttlingRateLimit": 10000.0,
            "cachingEnabled": false,
            "cacheTtlInSeconds": 300,
            "cacheDataEncrypted": false,
            "requireAuthorizationForCacheControl": true,
            "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"
        }
    },
    "tracingEnabled": false,
    "createdDate": "2022-07-18T10:11:18-07:00",
    "lastUpdatedDate": "2022-07-18T10:31:04-07:00"
}
```

Pour plus d'informations, consultez la section Configuration d'un stage pour une API REST dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateStageà la section Référence des AWS CLI commandes.

`update-usage-plan`

L'exemple de code suivant montre comment utiliser`update-usage-plan`.

AWS CLI

Pour modifier la période définie dans un plan d'utilisation

Commande :

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/period",value="MONTH"
```

Pour modifier la limite de quota définie dans un plan d'utilisation

Commande :

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/limit",value="500"
```

Pour modifier la limite de vitesse d'accélération définie dans un plan d'utilisation

Commande :

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/rateLimit",value="10"
```

Pour modifier la limite de rafale d'accélérateur définie dans un plan d'utilisation

Commande :

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/burstLimit",value="20"
```

- Pour plus de détails sur l'API, reportez-vous UpdateUsagePlanà la section Référence des AWS CLI commandes.

`update-usage`

L'exemple de code suivant montre comment utiliser`update-usage`.

AWS CLI

Pour modifier temporairement le quota d'une clé d'API pour la période en cours définie dans le plan d'utilisation

Commande :

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/
remaining",value="50"
```

- Pour plus de détails sur l'API, reportez-vous UpdateUsageà la section Référence des AWS CLI commandes.

API Gateway HTTP et exemples WebSocket d'API utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du protocole HTTP et de l'API AWS Command Line Interface with WebSocket API Gateway.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-api-mapping`

L'exemple de code suivant montre comment utiliser`create-api-mapping`.

AWS CLI

Pour créer un mappage d'API pour une API

L'`create-api-mapping`exemple suivant fait correspondre le `test` stage d'une API au `/myApi` chemin du nom de domaine `regional.example.com` personnalisé.

```
aws apigatewayv2 create-api-mapping \
    --domain-name regional.example.com \
    --api-mapping-key myApi \
    --api-id a1b2c3d4 \
    --stage test
```

Sortie :

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "0qzs2sy7bh",
    "ApiMappingKey": "myApi"
    "Stage": "test"
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateApiMappingà la section Référence des AWS CLI commandes.

`create-api`

L'exemple de code suivant montre comment utiliser`create-api`.

AWS CLI

Pour créer une API HTTP

L'`create-api`exemple suivant crée une API HTTP à l'aide de la création rapide. Vous pouvez utiliser la création rapide pour créer une API avec une intégration AWS Lambda ou HTTP, une route fourre-tout par défaut et une étape par défaut configurée pour déployer automatiquement les modifications. La commande suivante utilise la création rapide pour créer une API HTTP qui s'intègre à une fonction Lambda.

```
aws apigatewayv2 create-api \
    --name my-http-api \
    --protocol-type HTTP \
    --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

Sortie :

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T19:05:45+00:00",
    "Name": "my-http-api",
    "ProtocolType": "HTTP",
```

```
    "RouteSelectionExpression": "$request.method $request.path"
}
```

Pour plus d'informations, consultez la section Développement d'une API HTTP dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Pour créer une WebSocket API

L'`create-api`exemple suivant crée une WebSocket API portant le nom spécifié.

```
aws apigatewayv2 create-api \
    --name "myWebSocketApi" \
    --protocol-type WEBSOCKET \
    --route-selection-expression '$request.body.action'
```

Sortie :

```
{
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "Name": "myWebSocketApi",
    "CreatedDate": "2018-11-15T06:23:51Z",
    "ProtocolType": "WEBSOCKET",
    "RouteSelectionExpression": "'$request.body.action'",
    "ApiId": "aabbccddee"
}
```

Pour plus d'informations, consultez la section Créer une WebSocket API dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateApià la section Référence des AWS CLI commandes.

`create-authorizer`

L'exemple de code suivant montre comment utiliser`create-authorizer`.

AWS CLI

Pour créer un autorisateur JWT pour une API HTTP

L'`create-authorizer`exemple suivant crée un autorisateur JWT qui utilise Amazon Cognito comme fournisseur d'identité.

```
aws apigatewayv2 create-authorizer \
    --name my-jwt-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-type JWT \
    --identity-source '$request.header.Authorization' \
    --jwt-configuration Audience=123456abc,Issuer=https://cognito-idp.us-
west-2.amazonaws.com/us-west-2_abc123
```

Sortie :

```
{
    "AuthorizerId": "a1b2c3",
    "AuthorizerType": "JWT",
    "IdentitySource": [
        "$request.header.Authorization"
    ],
    "JwtConfiguration": {
        "Audience": [
            "123456abc"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
    },
    "Name": "my-jwt-authorizer"
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès au protocole HTTP APIs avec les autorisateurs JWT dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateAuthorizerà la section Référence des AWS CLI commandes.

`create-deployment`

L'exemple de code suivant montre comment utiliser`create-deployment`.

AWS CLI

Pour créer un déploiement pour une API

L'`create-deployment`exemple suivant crée un déploiement pour une API et associe ce déploiement à l'`dev`étape de l'API.

```
aws apigatewayv2 create-deployment \
    --api-id a1b2c3d4 \
    --stage-name dev
```

Sortie :

```
{
    "AutoDeployed": false,
    "CreatedDate": "2020-04-06T23:38:08Z",
    "DeploymentId": "53lz9l",
    "DeploymentStatus": "DEPLOYED"
}
```

Pour plus d'informations, consultez la section Déploiement d'API dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentà la section Référence des AWS CLI commandes.

`create-domain-name`

L'exemple de code suivant montre comment utiliser`create-domain-name`.

AWS CLI

Pour créer un nom de domaine personnalisé

L'`create-domain-name`exemple suivant crée un nom de domaine personnalisé régional pour une API.

```
aws apigatewayv2 create-domain-name \
    --domain-name regional.example.com \
    --domain-name-configurations CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

Sortie :

```
{
    "ApiMappingSelectionExpression": "$request.basepath",
    "DomainName": "regional.example.com",
```

```
    "DomainNameConfigurations": [
        {
            "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
            "EndpointType": "REGIONAL",
            "HostedZoneId": "123456789111",
            "SecurityPolicy": "TLS_1_2",
            "DomainNameStatus": "AVAILABLE"
        }
    ]
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDomainNameà la section Référence des AWS CLI commandes.

`create-integration`

L'exemple de code suivant montre comment utiliser`create-integration`.

AWS CLI

Pour créer une intégration WebSocket d'API

L'`create-integration`exemple suivant crée une intégration fictive pour une WebSocket API.

```
aws apigatewayv2 create-integration \
    --api-id aabbccddee \
    --passthrough-behavior WHEN_NO_MATCH \
    --timeout-in-millis 29000 \
    --connection-type INTERNET \
    --integration-type MOCK
```

Sortie :

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "0abcdef",
    "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
```

```
    "IntegrationType": "MOCK",
    "PassthroughBehavior": "WHEN_NO_MATCH",
    "PayloadFormatVersion": "1.0",
    "TimeoutInMillis": 29000
}
```

Pour plus d'informations, consultez Configurer une demande d'intégration d' WebSocket API dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

Pour créer une intégration d'API HTTP

L'`create-integration`exemple suivant crée une intégration AWS Lambda pour une API HTTP.

```
aws apigatewayv2 create-integration \
    --api-id a1b2c3d4 \
    --integration-type AWS_PROXY \
    --integration-uri arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --payload-format-version 2.0
```

Sortie :

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "0abcdef",
    "IntegrationMethod": "POST",
    "IntegrationType": "AWS_PROXY",
    "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "PayloadFormatVersion": "2.0",
    "TimeoutInMillis": 30000
}
```

Pour plus d'informations, consultez la section Configuration des intégrations pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateIntegrationà la section Référence des AWS CLI commandes.

`create-route`

L'exemple de code suivant montre comment utiliser`create-route`.

AWS CLI

Pour créer une route $default pour une WebSocket ou une API HTTP

L'`create-route`exemple suivant crée un `$default` itinéraire pour une API WebSocket ou HTTP.

```
aws apigatewayv2 create-route \
    --api-id aabbccddee \
    --route-key '$default'
```

Sortie :

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteKey": "$default",
    "RouteId": "1122334"
}
```

Pour plus d'informations, consultez la section Travailler avec des routes pour WebSocket APIs dans le manuel Amazon API Gateway Developer Guide.

Pour créer un itinéraire pour une API HTTP

L'`create-route`exemple suivant crée une route nommée `signup` qui accepte les requêtes POST.

```
aws apigatewayv2 create-route \
    --api-id aabbccddee \
    --route-key 'POST /signup'
```

Sortie :

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteKey": "POST /signup",
    "RouteId": "1122334"
}
```

Pour plus d'informations, consultez la section Travailler avec des routes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide

- Pour plus de détails sur l'API, reportez-vous CreateRouteà la section Référence des AWS CLI commandes.

`create-routing-rule`

L'exemple de code suivant montre comment utiliser`create-routing-rule`.

AWS CLI

Pour créer une règle de routage

L'`create-routing-rule`exemple suivant crée une règle de routage avec une priorité de`50`.

```
aws apigatewayv2 create-routing-rule \
    --domain-name 'regional.example.com' \
    --priority 50 \
    --conditions '[ \
        { \
            "MatchBasePaths": { \
                "AnyOf": [ \
                    "PetStoreShopper" \
                ] \
            } \
        } \
    ]' \
    --actions '[ \
        { \
            "InvokeApi": { \
                "ApiId": "abcd1234", \
                "Stage": "prod" \
            } \
        } \
    ]'
```

Sortie :

```
{
    "Actions": [
        {
```

```
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
    ],
    "Priority": 50,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

Pour plus d'informations, consultez la section Règles de routage pour connecter les étapes d'API à un nom de domaine personnalisé pour REST APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateRoutingRuleà la section Référence des AWS CLI commandes.

`create-stage`

L'exemple de code suivant montre comment utiliser`create-stage`.

AWS CLI

Pour créer une scène

L'`create-stage`exemple suivant crée une étape nommée dev pour une API.

```
aws apigatewayv2 create-stage \
    --api-id a1b2c3d4 \
    --stage-name dev
```

Sortie :

```
{
    "CreatedDate": "2020-04-06T23:23:46Z",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "LastUpdatedDate": "2020-04-06T23:23:46Z",
    "RouteSettings": {},
    "StageName": "dev",
    "StageVariables": {},
    "Tags": {}
}
```

Pour plus d'informations, consultez la section Travailler avec des étapes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateStageà la section Référence des AWS CLI commandes.

`create-vpc-link`

L'exemple de code suivant montre comment utiliser`create-vpc-link`.

AWS CLI

Pour créer un lien VPC pour une API HTTP

L'`create-vpc-link`exemple suivant crée un lien VPC pour HTTP. APIs

```
aws apigatewayv2 create-vpc-link \
    --name MyVpcLink \
    --subnet-ids subnet-aaaa subnet-bbbb \
    --security-group-ids sg1234 sg5678
```

Sortie :

```
{
    "CreatedDate": "2020-04-07T00:11:46Z",
    "Name": "MyVpcLink",
    "SecurityGroupIds": [
        "sg1234",
```

```
        "sg5678"
    ],
    "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
    ],
    "Tags": {},
    "VpcLinkId": "abcd123",
    "VpcLinkStatus": "PENDING",
    "VpcLinkStatusMessage": "VPC link is provisioning ENIs",
    "VpcLinkVersion": "V2"
}
```

Pour plus d'informations, consultez la section Utilisation de liens VPC pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateVpcLinkà la section Référence des AWS CLI commandes.

`delete-access-log-settings`

L'exemple de code suivant montre comment utiliser`delete-access-log-settings`.

AWS CLI

Pour désactiver la journalisation des accès pour une API

L'`delete-access-log-settings`exemple suivant supprime les paramètres du journal d'accès pour l'`$default`étape d'une API. Pour désactiver la journalisation des accès pour une étape, supprimez ses paramètres de journal d'accès.

```
aws apigatewayv2 delete-access-log-settings \
    --api-id a1b2c3d4 \
    --stage-name '$default'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration de la journalisation pour une API HTTP dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteAccessLogSettingsà la section Référence des AWS CLI commandes.

`delete-api-mapping`

L'exemple de code suivant montre comment utiliser`delete-api-mapping`.

AWS CLI

Pour supprimer un mappage d'API

L'`delete-api-mapping`exemple suivant supprime un mappage d'API pour le nom de domaine `api.example.com` personnalisé.

```
aws apigatewayv2 delete-api-mapping \
    --api-mapping-id a1b2c3 \
    --domain-name api.example.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteApiMappingà la section Référence des AWS CLI commandes.

`delete-api`

L'exemple de code suivant montre comment utiliser`delete-api`.

AWS CLI

Pour supprimer une API

L'`delete-api`exemple suivant supprime une API.

```
aws apigatewayv2 delete-api \
    --api-id a1b2c3d4
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les sections Working with HTTP APIs et Working with WebSocket APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteApià la section Référence des AWS CLI commandes.

`delete-authorizer`

L'exemple de code suivant montre comment utiliser`delete-authorizer`.

AWS CLI

Pour supprimer un autorisateur

L'`delete-authorizer`exemple suivant supprime un autorisateur.

```
aws apigatewayv2 delete-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contrôle de l'accès au protocole HTTP APIs avec les autorisateurs JWT dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteAuthorizerà la section Référence des AWS CLI commandes.

`delete-cors-configuration`

L'exemple de code suivant montre comment utiliser`delete-cors-configuration`.

AWS CLI

Pour supprimer la configuration CORS d'une API HTTP

L'`delete-cors-configuration`exemple suivant désactive le CORS pour une API HTTP en supprimant sa configuration CORS.

```
aws apigatewayv2 delete-cors-configuration \
    --api-id a1b2c3d4
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration de CORS pour une API HTTP dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteCorsConfigurationà la section Référence des AWS CLI commandes.

`delete-deployment`

L'exemple de code suivant montre comment utiliser`delete-deployment`.

AWS CLI

Pour supprimer un déploiement

L'`delete-deployment`exemple suivant supprime le déploiement d'une API.

```
aws apigatewayv2 delete-deployment \
    --api-id a1b2c3d4 \
    --deployment-id a1b2c3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Déploiement d'API dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDeploymentà la section Référence des AWS CLI commandes.

`delete-domain-name`

L'exemple de code suivant montre comment utiliser`delete-domain-name`.

AWS CLI

Pour supprimer un nom de domaine personnalisé

L'`delete-domain-name`exemple suivant supprime un nom de domaine personnalisé.

```
aws apigatewayv2 delete-domain-name \
    --domain-name api.example.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDomainNameà la section Référence des AWS CLI commandes.

`delete-integration`

L'exemple de code suivant montre comment utiliser`delete-integration`.

AWS CLI

Pour supprimer une intégration

L'`delete-integration`exemple suivant supprime une intégration d'API.

```
aws apigatewayv2 delete-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les sections Configuration des intégrations pour le protocole HTTP APIs et Configuration des intégrations d' WebSocket API dans le manuel du développeur Amazon API Gateway.

- Pour plus de détails sur l'API, reportez-vous DeleteIntegrationà la section Référence des AWS CLI commandes.

`delete-route-settings`

L'exemple de code suivant montre comment utiliser`delete-route-settings`.

AWS CLI

Pour supprimer les paramètres d'itinéraire

L'`delete-route-settings`exemple suivant supprime les paramètres d'itinéraire pour l'itinéraire spécifié.

```
aws apigatewayv2 delete-route-settings \
    --api-id a1b2c3d4 \
    --stage-name dev \
    --route-key 'GET /pets'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec des routes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteRouteSettingsà la section Référence des AWS CLI commandes.

`delete-route`

L'exemple de code suivant montre comment utiliser`delete-route`.

AWS CLI

Pour supprimer un itinéraire

L'`delete-route`exemple suivant supprime une route d'API.

```
aws apigatewayv2 delete-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec des routes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteRouteà la section Référence des AWS CLI commandes.

`delete-routing-rule`

L'exemple de code suivant montre comment utiliser`delete-routing-rule`.

AWS CLI

Pour supprimer une règle de routage

L'`delete-routing-rule`exemple suivant supprime une règle de routage pour un nom de domaine personnalisé.

```
aws apigatewayv2 delete-routing-rule \
    --domain-name 'regional.example.com' \
    --routing-rule-id aaa111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Règles de routage pour connecter les étapes d'API à un nom de domaine personnalisé pour REST APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteRoutingRuleà la section Référence des AWS CLI commandes.

`delete-stage`

L'exemple de code suivant montre comment utiliser`delete-stage`.

AWS CLI

Pour supprimer une étape

L'`delete-stage`exemple suivant supprime le `test` stage d'une API.

```
aws apigatewayv2 delete-stage \
    --api-id a1b2c3d4 \
    --stage-name test
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec des étapes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteStageà la section Référence des AWS CLI commandes.

`delete-vpc-link`

L'exemple de code suivant montre comment utiliser`delete-vpc-link`.

AWS CLI

Pour supprimer un lien VPC pour une API HTTP

L'`delete-vpc-link`exemple suivant supprime un lien VPC.

```
aws apigatewayv2 delete-vpc-link \
    --vpc-link-id abcd123
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de liens VPC pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteVpcLinkà la section Référence des AWS CLI commandes.

`export-api`

L'exemple de code suivant montre comment utiliser`export-api`.

AWS CLI

Pour exporter la définition OpenAPI d'une API HTTP

L'`export-api`exemple suivant exporte une définition OpenAPI 3.0 d'un stage d'API nommé `prod` vers un fichier YAML nommé. `stage-definition.yaml` Le fichier de définition exporté inclut les extensions API Gateway par défaut.

```
aws apigatewayv2 export-api \
    --api-id a1b2c3d4 \
    --output-type YAML \
    --specification OAS30 \
    --stage-name prod \
    stage-definition.yaml
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Exporter une API HTTP depuis API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ExportApià la section Référence des AWS CLI commandes.

`get-api-mapping`

L'exemple de code suivant montre comment utiliser`get-api-mapping`.

AWS CLI

Pour obtenir des informations sur un mappage d'API pour un nom de domaine personnalisé

L'`get-api-mapping`exemple suivant affiche des informations sur un mappage d'API pour le nom de domaine `api.example.com` personnalisé.

```
aws apigatewayv2 get-api-mapping \
    --api-mapping-id a1b2c3 \
    --domain-name api.example.com
```

Sortie :

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "a1b2c3d5",
    "ApiMappingKey": "myTestApi"
    "Stage": "test"
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetApiMappingà la section Référence des AWS CLI commandes.

`get-api-mappings`

L'exemple de code suivant montre comment utiliser`get-api-mappings`.

AWS CLI

Pour obtenir des mappages d'API pour un nom de domaine personnalisé

L'`get-api-mappings`exemple suivant affiche une liste de tous les mappages d'API pour le nom de domaine `api.example.com` personnalisé.

```
aws apigatewayv2 get-api-mappings \
    --domain-name api.example.com
```

Sortie :

```
{
    "Items": [
```

```
        {
            "ApiId": "a1b2c3d4",
            "ApiMappingId": "a1b2c3d5",
            "ApiMappingKey": "myTestApi"
            "Stage": "test"
        },
        {
            "ApiId": "a5b6c7d8",
            "ApiMappingId": "a1b2c3d6",
            "ApiMappingKey": "myDevApi"
            "Stage": "dev"
        },
    ]
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetApiMappingsà la section Référence des AWS CLI commandes.

`get-api`

L'exemple de code suivant montre comment utiliser`get-api`.

AWS CLI

Pour récupérer des informations sur une API

L'`get-api`exemple suivant affiche des informations sur une API.

```
aws apigatewayv2 get-api \
    --api-id a1b2c3d4
```

Sortie :

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-03-28T00:32:37Z",
    "Name": "my-api",
```

```
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {
        "department": "finance"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetApià la section Référence des AWS CLI commandes.

`get-apis`

L'exemple de code suivant montre comment utiliser`get-apis`.

AWS CLI

Pour récupérer une liste de APIs

L'`get-apis`exemple suivant répertorie toutes les informations APIs pour l'utilisateur actuel.

```
aws apigatewayv2 get-apis
```

Sortie :

```
{
    "Items": [
        {
            "ApiEndpoint": "wss://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
            "ApiId": "a1b2c3d4",
            "ApiKeySelectionExpression": "$request.header.x-api-key",
            "CreatedDate": "2020-04-07T20:21:59Z",
            "Name": "my-websocket-api",
            "ProtocolType": "WEBSOCKET",
            "RouteSelectionExpression": "$request.body.message",
            "Tags": {}
        },
        {
            "ApiEndpoint": "https://a1b2c3d5.execute-api.us-west-2.amazonaws.com",
            "ApiId": "a1b2c3d5",
            "ApiKeySelectionExpression": "$request.header.x-api-key",
            "CreatedDate": "2020-04-07T20:23:50Z",
            "Name": "my-http-api",
```

```
            "ProtocolType": "HTTP",
            "RouteSelectionExpression": "$request.method $request.path",
            "Tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez les sections Working with HTTP APIs et Working with WebSocket APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetApisà la section Référence des AWS CLI commandes.

`get-authorizer`

L'exemple de code suivant montre comment utiliser`get-authorizer`.

AWS CLI

Pour récupérer des informations sur un autorisateur

L'`get-authorizer`exemple suivant affiche des informations sur un autorisateur.

```
aws apigatewayv2 get-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3
```

Sortie :

```
{
    "AuthorizerId": "a1b2c3",
    "AuthorizerType": "JWT",
    "IdentitySource": [
        "$request.header.Authorization"
    ],
    "JwtConfiguration": {
        "Audience": [
            "123456abc"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
    },
    "Name": "my-jwt-authorizer"
```

```
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès au protocole HTTP APIs avec les autorisateurs JWT dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetAuthorizerà la section Référence des AWS CLI commandes.

`get-authorizers`

L'exemple de code suivant montre comment utiliser`get-authorizers`.

AWS CLI

Pour récupérer la liste des autorisateurs pour une API

L'`get-authorizers`exemple suivant affiche la liste de tous les autorisateurs pour une API.

```
aws apigatewayv2 get-authorizers \
    --api-id a1b2c3d4
```

Sortie :

```
{
    "Items": [
        {
            "AuthorizerId": "a1b2c3",
            "AuthorizerType": "JWT",
            "IdentitySource": [
                "$request.header.Authorization"
            ],
            "JwtConfiguration": {
                "Audience": [
                    "123456abc"
                ],
                "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_abc123"
            },
            "Name": "my-jwt-authorizer"
        },
        {
            "AuthorizerId": "a1b2c4",
```

```
            "AuthorizerType": "JWT",
            "IdentitySource": [
                "$request.header.Authorization"
            ],
            "JwtConfiguration": {
                "Audience": [
                    "6789abcde"
                ],
                "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_abc234"
            },
            "Name": "new-jwt-authorizer"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès au protocole HTTP APIs avec les autorisateurs JWT dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetAuthorizersà la section Référence des AWS CLI commandes.

`get-deployment`

L'exemple de code suivant montre comment utiliser`get-deployment`.

AWS CLI

Pour récupérer des informations relatives à un déploiement

L'`get-deployment`exemple suivant affiche les informations relatives à un déploiement.

```
aws apigatewayv2 get-deployment \
    --api-id a1b2c3d4 \
    --deployment-id abcdef
```

Sortie :

```
{
    "AutoDeployed": true,
    "CreatedDate": "2020-04-07T23:58:40Z",
    "DeploymentId": "abcdef",
```

```
    "DeploymentStatus": "DEPLOYED",
    "Description": "Automatic deployment triggered by changes to the Api
 configuration"
}
```

Pour plus d'informations, consultez la section Déploiement d'API dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDeploymentà la section Référence des AWS CLI commandes.

`get-deployments`

L'exemple de code suivant montre comment utiliser`get-deployments`.

AWS CLI

Pour récupérer la liste des déploiements

L'`get-deployments`exemple suivant affiche la liste de tous les déploiements d'une API.

```
aws apigatewayv2 get-deployments \
    --api-id a1b2c3d4
```

Sortie :

```
{
    "Items": [
        {
            "AutoDeployed": true,
            "CreatedDate": "2020-04-07T23:58:40Z",
            "DeploymentId": "abcdef",
            "DeploymentStatus": "DEPLOYED",
            "Description": "Automatic deployment triggered by changes to the Api
 configuration"
        },
        {
            "AutoDeployed": true,
            "CreatedDate": "2020-04-06T00:33:00Z",
            "DeploymentId": "bcdefg",
            "DeploymentStatus": "DEPLOYED",
            "Description": "Automatic deployment triggered by changes to the Api
 configuration"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Déploiement d'API dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDeploymentsà la section Référence des AWS CLI commandes.

`get-domain-name`

L'exemple de code suivant montre comment utiliser`get-domain-name`.

AWS CLI

Pour récupérer des informations sur un nom de domaine personnalisé

L'`get-domain-name`exemple suivant affiche les informations relatives à un nom de domaine personnalisé.

```
aws apigatewayv2 get-domain-name \
    --domain-name api.example.com
```

Sortie :

```
{
    "ApiMappingSelectionExpression": "$request.basepath",
    "DomainName": "api.example.com",
    "DomainNameConfigurations": [
        {
            "ApiGatewayDomainName": "d-1234.execute-api.us-west-2.amazonaws.com",
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
            "EndpointType": "REGIONAL",
            "HostedZoneId": "123456789111",
            "SecurityPolicy": "TLS_1_2",
            "DomainNameStatus": "AVAILABLE"
        }
    ],
    "Tags": {}
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDomainNameà la section Référence des AWS CLI commandes.

`get-domain-names`

L'exemple de code suivant montre comment utiliser`get-domain-names`.

AWS CLI

Pour récupérer une liste de noms de domaine personnalisés

L'`get-domain-names`exemple suivant affiche une liste de tous les noms de domaine personnalisés pour l'utilisateur actuel.

```
aws apigatewayv2 get-domain-names
```

Sortie :

```
{
    "Items": [
        {
            "ApiMappingSelectionExpression": "$request.basepath",
            "DomainName": "api.example.com",
            "DomainNameConfigurations": [
                {
                    "ApiGatewayDomainName": "d-1234.execute-api.us-
west-2.amazonaws.com",
                    "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
                    "EndpointType": "REGIONAL",
                    "HostedZoneId": "123456789111",
                    "SecurityPolicy": "TLS_1_2",
                    "DomainNameStatus": "AVAILABLE"
                }
            ]
        },
        {
            "ApiMappingSelectionExpression": "$request.basepath",
            "DomainName": "newApi.example.com",
```

```
            "DomainNameConfigurations": [
                {
                    "ApiGatewayDomainName": "d-5678.execute-api.us-
west-2.amazonaws.com",
                    "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
                    "EndpointType": "REGIONAL",
                    "HostedZoneId": "123456789222",
                    "SecurityPolicy": "TLS_1_2",
                    "DomainNameStatus": "AVAILABLE"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDomainNamesà la section Référence des AWS CLI commandes.

`get-integration`

L'exemple de code suivant montre comment utiliser`get-integration`.

AWS CLI

Pour récupérer des informations sur une intégration

L'`get-integration`exemple suivant affiche les informations relatives à une intégration.

```
aws apigatewayv2 get-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3
```

Sortie :

```
{
    "ApiGatewayManaged": true,
    "ConnectionType": "INTERNET",
    "IntegrationId": "a1b2c3",
```

```
    "IntegrationMethod": "POST",
    "IntegrationType": "AWS_PROXY",
    "IntegrationUri": "arn:aws:lambda:us-west-2:12356789012:function:hello12",
    "PayloadFormatVersion": "2.0",
    "TimeoutInMillis": 30000
}
```

Pour plus d'informations, consultez les sections Configuration des intégrations pour le protocole HTTP APIs et Configuration des intégrations d' WebSocket API dans le manuel du développeur Amazon API Gateway.

- Pour plus de détails sur l'API, reportez-vous GetIntegrationà la section Référence des AWS CLI commandes.

`get-integrations`

L'exemple de code suivant montre comment utiliser`get-integrations`.

AWS CLI

Pour récupérer une liste d'intégrations

L'`get-integrations`exemple suivant affiche la liste de toutes les intégrations d'une API.

```
aws apigatewayv2 get-integrations \
    --api-id a1b2c3d4
```

Sortie :

```
{
    "Items": [
        {
            "ApiGatewayManaged": true,
            "ConnectionType": "INTERNET",
            "IntegrationId": "a1b2c3",
            "IntegrationMethod": "POST",
            "IntegrationType": "AWS_PROXY",
            "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
            "PayloadFormatVersion": "2.0",
            "TimeoutInMillis": 30000
```

```
        },
        {
            "ConnectionType": "INTERNET",
            "IntegrationId": "a1b2c4",
            "IntegrationMethod": "ANY",
            "IntegrationType": "HTTP_PROXY",
            "IntegrationUri": "https://www.example.com",
            "PayloadFormatVersion": "1.0",
            "TimeoutInMillis": 30000
        }
    ]
}
```

Pour plus d'informations, consultez les sections Configuration des intégrations pour le protocole HTTP APIs et Configuration des intégrations d' WebSocket API dans le manuel du développeur Amazon API Gateway.

- Pour plus de détails sur l'API, reportez-vous GetIntegrationsà la section Référence des AWS CLI commandes.

`get-route`

L'exemple de code suivant montre comment utiliser`get-route`.

AWS CLI

Pour récupérer des informations sur un itinéraire

L'`get-route`exemple suivant affiche les informations relatives à un itinéraire.

```
aws apigatewayv2 get-route \
    --api-id a1b2c3d4 \
    --route-id 72jz1wk
```

Sortie :

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteId": "72jz1wk",
    "RouteKey": "ANY /pets",
    "Target": "integrations/a1b2c3"
```

```
}
```

Pour plus d'informations, consultez la section Travailler avec des routes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide

- Pour plus de détails sur l'API, reportez-vous GetRouteà la section Référence des AWS CLI commandes.

`get-routes`

L'exemple de code suivant montre comment utiliser`get-routes`.

AWS CLI

Pour récupérer une liste d'itinéraires

L'`get-routes`exemple suivant affiche une liste de toutes les routes d'une API.

```
aws apigatewayv2 get-routes \
    --api-id a1b2c3d4
```

Sortie :

```
{
    "Items": [
        {
            "ApiKeyRequired": false,
            "AuthorizationType": "NONE",
            "RouteId": "72jz1wk",
            "RouteKey": "ANY /admin",
            "Target": "integrations/a1b2c3"
        },
        {
            "ApiGatewayManaged": true,
            "ApiKeyRequired": false,
            "AuthorizationType": "NONE",
            "RouteId": "go65gqi",
            "RouteKey": "$default",
            "Target": "integrations/a1b2c4"
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des routes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide

- Pour plus de détails sur l'API, reportez-vous GetRoutesà la section Référence des AWS CLI commandes.

`get-routing-rule`

L'exemple de code suivant montre comment utiliser`get-routing-rule`.

AWS CLI

Pour obtenir une règle de routage

L'`get-routing-rule`exemple suivant permet d'obtenir une règle de routage pour un nom de domaine.

```
aws apigatewayv2 get-routing-rule \
    --domain-name 'regional.example.com' \
    --routing-rule-id aaa111
```

Sortie :

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
    ],
```

```
    "Priority": 50,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

Pour plus d'informations, consultez la section Règles de routage pour connecter les étapes d'API à un nom de domaine personnalisé pour REST APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetRoutingRuleà la section Référence des AWS CLI commandes.

`get-stage`

L'exemple de code suivant montre comment utiliser`get-stage`.

AWS CLI

Pour récupérer des informations sur une étape

L'`get-stage`exemple suivant affiche des informations sur le `prod` stade d'une API.

```
aws apigatewayv2 get-stage \
    --api-id a1b2c3d4 \
    --stage-name prod
```

Sortie :

```
{
    "CreatedDate": "2020-04-08T00:36:05Z",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "DeploymentId": "x1zwyv",
    "LastUpdatedDate": "2020-04-08T00:36:13Z",
    "RouteSettings": {},
    "StageName": "prod",
    "StageVariables": {
        "function": "my-prod-function"
    },
    "Tags": {}
```

```
}
```

Pour plus d'informations, consultez la section Travailler avec des étapes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetStageà la section Référence des AWS CLI commandes.

`get-stages`

L'exemple de code suivant montre comment utiliser`get-stages`.

AWS CLI

Pour récupérer la liste des étapes

L'`get-stages`exemple suivant répertorie toutes les étapes d'une API.

```
aws apigatewayv2 get-stages \
    --api-id a1b2c3d4
```

Sortie :

```
{
    "Items": [
        {
            "ApiGatewayManaged": true,
            "AutoDeploy": true,
            "CreatedDate": "2020-04-08T00:08:44Z",
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "DeploymentId": "dty748",
            "LastDeploymentStatusMessage": "Successfully deployed stage with
 deployment ID 'dty748'",
            "LastUpdatedDate": "2020-04-08T00:09:49Z",
            "RouteSettings": {},
            "StageName": "$default",
            "StageVariables": {},
            "Tags": {}
        },
        {
```

```
            "AutoDeploy": true,
            "CreatedDate": "2020-04-08T00:35:06Z",
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "LastUpdatedDate": "2020-04-08T00:35:48Z",
            "RouteSettings": {},
            "StageName": "dev",
            "StageVariables": {
                "function": "my-dev-function"
            },
            "Tags": {}
        },
        {
            "CreatedDate": "2020-04-08T00:36:05Z",
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "DeploymentId": "x1zwyv",
            "LastUpdatedDate": "2020-04-08T00:36:13Z",
            "RouteSettings": {},
            "StageName": "prod",
            "StageVariables": {
                "function": "my-prod-function"
            },
            "Tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des étapes pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetStagesà la section Référence des AWS CLI commandes.

`get-tags`

L'exemple de code suivant montre comment utiliser`get-tags`.

AWS CLI

Pour récupérer une liste de balises pour une ressource

L'`get-tags`exemple suivant répertorie toutes les balises d'une API.

```
aws apigatewayv2 get-tags \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4
```

Sortie :

```
{
    "Tags": {
        "owner": "dev-team",
        "environment": "prod"
    }
}
```

Pour plus d'informations, consultez la section Marquage de vos ressources API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetTagsà la section Référence des AWS CLI commandes.

`get-vpc-link`

L'exemple de code suivant montre comment utiliser`get-vpc-link`.

AWS CLI

Pour récupérer des informations sur un lien VPC

L'`get-vpc-link`exemple suivant affiche des informations sur un lien VPC.

```
aws apigatewayv2 get-vpc-link \
    --vpc-link-id abcd123
```

Sortie :

```
{
    "CreatedDate": "2020-04-07T00:27:47Z",
    "Name": "MyVpcLink",
    "SecurityGroupIds": [
        "sg1234",
```

```
        "sg5678"
    ],
    "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
    ],
    "Tags": {},
    "VpcLinkId": "abcd123",
    "VpcLinkStatus": "AVAILABLE",
    "VpcLinkStatusMessage": "VPC link is ready to route traffic",
    "VpcLinkVersion": "V2"
}
```

Pour plus d'informations, consultez la section Utilisation de liens VPC pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetVpcLinkà la section Référence des AWS CLI commandes.

`get-vpc-links`

L'exemple de code suivant montre comment utiliser`get-vpc-links`.

AWS CLI

Pour récupérer une liste de liens VPC

L'`get-vpc-links`exemple suivant affiche une liste de tous les liens VPC pour l'utilisateur actuel.

```
aws apigatewayv2 get-vpc-links
```

Sortie :

```
{
    "Items": [
        {
            "CreatedDate": "2020-04-07T00:27:47Z",
            "Name": "MyVpcLink",
            "SecurityGroupIds": [
                "sg1234",
                "sg5678"
            ],
```

```
            "SubnetIds": [
                "subnet-aaaa",
                "subnet-bbbb"
            ],
            "Tags": {},
            "VpcLinkId": "abcd123",
            "VpcLinkStatus": "AVAILABLE",
            "VpcLinkStatusMessage": "VPC link is ready to route traffic",
            "VpcLinkVersion": "V2"
        }
        {
            "CreatedDate": "2020-04-07T00:27:47Z",
            "Name": "MyOtherVpcLink",
            "SecurityGroupIds": [
                "sg1234",
                "sg5678"
            ],
            "SubnetIds": [
                "subnet-aaaa",
                "subnet-bbbb"
            ],
            "Tags": {},
            "VpcLinkId": "abcd456",
            "VpcLinkStatus": "AVAILABLE",
            "VpcLinkStatusMessage": "VPC link is ready to route traffic",
            "VpcLinkVersion": "V2"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de liens VPC pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetVpcLinksà la section Référence des AWS CLI commandes.

`import-api`

L'exemple de code suivant montre comment utiliser`import-api`.

AWS CLI

Pour importer une API HTTP

L'`import-api`exemple suivant crée une API HTTP à partir d'un fichier de définition OpenAPI 3.0 nommé. `api-definition.yaml`

```
aws apigatewayv2 import-api \
    --body file://api-definition.yaml
```

Contenu de `api-definition.yaml` :

```
openapi: 3.0.1
info:
    title: My Lambda API
    version: v1.0
paths:
    /hello:
        x-amazon-apigateway-any-method:
            x-amazon-apigateway-integration:
                payloadFormatVersion: 2.0
                type: aws_proxy
                httpMethod: POST
                uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123456789012:function:hello/invocations
                connectionType: INTERNET
```

Sortie :

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T17:19:38+00:00",
    "Name": "My Lambda API",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

Pour plus d'informations, consultez la section Travailler avec les définitions OpenAPI pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ImportApià la section Référence des AWS CLI commandes.

`list-routing-rules`

L'exemple de code suivant montre comment utiliser`list-routing-rules`.

AWS CLI

Pour répertorier les règles de routage

L'`list-routing-rules`exemple suivant répertorie les règles de routage pour un nom de domaine.

```
aws apigatewayv2 list-routing-rules \
    --domain-name 'regional.example.com'
```

Sortie :

```
{
    "RoutingRules": [
        {
            "Actions": [
                {
                    "InvokeApi": {
                        "ApiId": "abcd1234",
                        "Stage": "prod",
                        "StripBasePath": false
                    }
                }
            ],
            "Conditions": [
                {
                    "MatchBasePaths": {
                        "AnyOf": [
                            "PetStoreShopper"
                        ]
                    }
                }
            ],
            "Priority": 150,
            "RoutingRuleArn": "arn:aws:apigateway:us-east-1:123456789012:/
domainnames/regional.example.com/routingrules/aaa111",
            "RoutingRuleId": "aaa111"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Règles de routage pour connecter les étapes d'API à un nom de domaine personnalisé pour REST APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListRoutingRulesà la section Référence des AWS CLI commandes.

`put-routing-rule`

L'exemple de code suivant montre comment utiliser`put-routing-rule`.

AWS CLI

Pour mettre à jour une règle de routage

L'`put-routing-rule`exemple suivant met à jour la priorité d'une règle de routage.

```
aws apigatewayv2 put-routing-rule \
    --domain-name 'regional.example.com' \
    --priority 150 \
    --conditions '[ \
        { \
            "MatchBasePaths": { \
                "AnyOf": [ \
                    "PetStoreShopper" \
                ] \
            } \
        } \
    ]' \
    --actions '[ \
        { \
            "InvokeApi": { \
                "ApiId": "abcd1234", \
                "Stage": "prod" \
            } \
        } \
    ]'
```

Sortie :

```
{
```

```
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
    ],
    "Priority": 150,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

Pour plus d'informations, consultez la section Règles de routage pour connecter les étapes d'API à un nom de domaine personnalisé pour REST APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutRoutingRuleà la section Référence des AWS CLI commandes.

`reimport-api`

L'exemple de code suivant montre comment utiliser`reimport-api`.

AWS CLI

Pour réimporter une API HTTP

L'`reimport-api`exemple suivant met à jour une API HTTP existante afin d'utiliser la définition OpenAPI 3.0 spécifiée dans. `api-definition.yaml`

```
aws apigatewayv2 reimport-api \
```

```
    --body file://api-definition.yaml \
    --api-id a1b2c3d4
```

Contenu de `api-definition.yaml` :

```
openapi: 3.0.1
info:
    title: My Lambda API
    version: v1.0
paths:
    /hello:
        x-amazon-apigateway-any-method:
            x-amazon-apigateway-integration:
                payloadFormatVersion: 2.0
                type: aws_proxy
                httpMethod: POST
                uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:12356789012:function:hello/invocations
                connectionType: INTERNET
```

Sortie :

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T17:19:38+00:00",
    "Name": "My Lambda API",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

Pour plus d'informations, consultez la section Travailler avec les définitions OpenAPI pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ReimportApià la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute une balise avec le nom de la clé `Department` et une valeur de `Accounting` à l'API spécifiée.

```
aws apigatewayv2 tag-resource \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \
    --tags Department=Accounting
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`untag-resource`exemple suivant supprime les balises portant les noms des clés `Project` et `Owner` de l'API spécifiée.

```
aws apigatewayv2 untag-resource \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \
    --tag-keys Project Owner
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-api-mapping`

L'exemple de code suivant montre comment utiliser`update-api-mapping`.

AWS CLI

Pour mettre à jour un mappage d'API

L'`update-api-mapping`exemple suivant modifie un mappage d'API pour un nom de domaine personnalisé. Par conséquent, l'URL de base utilisant le nom de domaine personnalisé pour l'API et le stage spécifiés devient`https://api.example.com/dev`.

```
aws apigatewayv2 update-api-mapping \
    --api-id a1b2c3d4 \
    --stage dev \
    --domain-name api.example.com \
    --api-mapping-id 0qzs2sy7bh \
    --api-mapping-key dev
```

Sortie :

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "0qzs2sy7bh",
    "ApiMappingKey": "dev"
    "Stage": "dev"
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateApiMappingà la section Référence des AWS CLI commandes.

`update-api`

L'exemple de code suivant montre comment utiliser`update-api`.

AWS CLI

Pour activer CORS pour une API HTTP

L'`update-api`exemple suivant met à jour la configuration CORS de l'API spécifiée pour autoriser les demandes provenant de`https://www.example.com`.

```
aws apigatewayv2 update-api \
    --api-id a1b2c3d4 \
    --cors-configuration AllowOrigins=https://www.example.com
```

Sortie :

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CorsConfiguration": {
        "AllowCredentials": false,
        "AllowHeaders": [
            "header1",
            "header2"
        ],
        "AllowMethods": [
            "GET",
            "OPTIONS"
        ],
        "AllowOrigins": [
            "https://www.example.com"
        ]
    },
    "CreatedDate": "2020-04-08T18:39:37+00:00",
    "Name": "my-http-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

Pour plus d'informations, consultez la section Configuration de CORS pour une API HTTP dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateApià la section Référence des AWS CLI commandes.

`update-authorizer`

L'exemple de code suivant montre comment utiliser`update-authorizer`.

AWS CLI

Pour mettre à jour un autorisateur

L'`update-authorizer`exemple suivant remplace la source d'identité d'un autorisateur JWT par un en-tête nommé. `Authorization`

```
aws apigatewayv2 update-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3 \
    --identity-source '$request.header.Authorization'
```

Sortie :

```
{
    "AuthorizerId": "a1b2c3",
    "AuthorizerType": "JWT",
    "IdentitySource": [
        "$request.header.Authorization"
    ],
    "JwtConfiguration": {
        "Audience": [
            "123456abc"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
    },
    "Name": "my-jwt-authorizer"
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès au protocole HTTP APIs avec les autorisateurs JWT dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateAuthorizerà la section Référence des AWS CLI commandes.

update-deployment

L'exemple de code suivant montre comment utiliserupdate-deployment.

AWS CLI

Pour modifier la description d'un déploiement

L'update-deploymentexemple suivant met à jour la description d'un déploiement.

```
aws apigatewayv2 update-deployment \
    --api-id a1b2c3d4 \
    --deployment-id abcdef \
    --description 'Manual deployment to fix integration test failures.'
```

Sortie :

```
{
    "AutoDeployed": false,
    "CreatedDate": "2020-02-05T16:21:48+00:00",
    "DeploymentId": "abcdef",
    "DeploymentStatus": "DEPLOYED",
    "Description": "Manual deployment to fix integration test failures."
}
```

Pour plus d'informations, consultez la section Développement d'une API HTTP dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDeploymentà la section Référence des AWS CLI commandes.

update-domain-name

L'exemple de code suivant montre comment utiliserupdate-domain-name.

AWS CLI

Pour mettre à jour un nom de domaine personnalisé

L'update-domain-nameexemple suivant indique un nouveau certificat ACM pour le nom de domaine api.example.com personnalisé.

```
aws apigatewayv2 update-domain-name \
    --domain-name api.example.com \
    --domain-name-configurations CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

Sortie :

```
{
    "ApiMappingSelectionExpression": "$request.basepath",
    "DomainName": "regional.example.com",
    "DomainNameConfigurations": [
        {
            "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
            "EndpointType": "REGIONAL",
            "HostedZoneId": "123456789111",
            "SecurityPolicy": "TLS_1_2",
            "DomainNameStatus": "AVAILABLE"
        }
    ]
}
```

Pour plus d'informations, consultez Configuration d'un nom de domaine personnalisé régional dans API Gateway dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainNameà la section Référence des AWS CLI commandes.

`update-integration`

L'exemple de code suivant montre comment utiliser`update-integration`.

AWS CLI

Pour mettre à jour une intégration Lambda

L'`update-integration`exemple suivant met à jour une intégration AWS Lambda existante pour utiliser la fonction Lambda spécifiée.

```
aws apigatewayv2 update-integration \
```

```
    --api-id a1b2c3d4 \
    --integration-id a1b2c3 \
    --integration-uri arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123456789012:function:my-new-function/invocations
```

Sortie :

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "a1b2c3",
    "IntegrationMethod": "POST",
    "IntegrationType": "AWS_PROXY",
    "IntegrationUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123456789012:function:my-new-function/
invocations",
    "PayloadFormatVersion": "2.0",
    "TimeoutInMillis": 5000
}
```

Pour plus d'informations, consultez les sections Configuration des intégrations pour le protocole HTTP APIs et Configuration des intégrations d' WebSocket API dans le manuel du développeur Amazon API Gateway.

- Pour plus de détails sur l'API, reportez-vous UpdateIntegrationà la section Référence des AWS CLI commandes.

`update-route`

L'exemple de code suivant montre comment utiliser`update-route`.

AWS CLI

Exemple 1 : Pour mettre à jour l'intégration d'un itinéraire

L'`update-route`exemple suivant met à jour l'intégration d'un itinéraire spécifié.

```
aws apigatewayv2 update-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3 \
    --target integrations/a1b2c6
```

Sortie :

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteId": "a1b2c3",
    "RouteKey": "ANY /pets",
    "Target": "integrations/a1b2c6"
}
```

Exemple 2 : pour ajouter un autorisateur à un itinéraire

L'`update-route`exemple suivant met à jour l'itinéraire spécifié pour utiliser un autorisateur JWT.

```
aws apigatewayv2 update-route \
    --api-id a1b2c3d4  \
    --route-id a1b2c3  \
    --authorization-type JWT \
    --authorizer-id a1b2c5 \
    --authorization-scopes user.id user.email
```

Sortie :

```
{
    "ApiKeyRequired": false,
    "AuthorizationScopes": [
        "user.id",
        "user.email"
    ],
    "AuthorizationType": "JWT",
    "AuthorizerId": "a1b2c5",
    "OperationName": "GET HTTP",
    "RequestParameters": {},
    "RouteId": "a1b2c3",
    "RouteKey": "GET /pets",
    "Target": "integrations/a1b2c6"
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès au protocole HTTP APIs avec les autorisateurs JWT dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateRouteà la section Référence des AWS CLI commandes.

update-stage

L'exemple de code suivant montre comment utiliserupdate-stage.

AWS CLI

Pour configurer la régulation personnalisée

L'update-stageexemple suivant configure une régulation personnalisée pour l'étape et l'itinéraire spécifiés d'une API.

```
aws apigatewayv2 update-stage \
    --api-id a1b2c3d4 \
    --stage-name dev \
    --route-settings '{"GET /pets":
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

Sortie :

```
{
    "CreatedDate": "2020-04-05T16:21:16+00:00",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "DeploymentId": "shktxb",
    "LastUpdatedDate": "2020-04-08T22:23:17+00:00",
    "RouteSettings": {
        "GET /pets": {
            "ThrottlingBurstLimit": 100,
            "ThrottlingRateLimit": 2000.0
        }
    },
    "StageName": "dev",
    "StageVariables": {},
    "Tags": {}
}
```

Pour plus d'informations, consultez la section Protection de votre API HTTP dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateStageà la section Référence des AWS CLI commandes.

update-vpc-link

L'exemple de code suivant montre comment utiliserupdate-vpc-link.

AWS CLI

Pour mettre à jour un lien VPC

L'update-vpc-linkexemple suivant met à jour le nom d'un lien VPC. Une fois que vous avez créé un lien VPC, vous ne pouvez pas modifier ses groupes de sécurité ou ses sous-réseaux.

```
aws apigatewayv2 update-vpc-link \
    --vpc-link-id abcd123 \
    --name MyUpdatedVpcLink
```

Sortie :

```
{
    "CreatedDate": "2020-04-07T00:27:47Z",
    "Name": "MyUpdatedVpcLink",
    "SecurityGroupIds": [
        "sg1234",
        "sg5678"
    ],
    "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
    ],
    "Tags": {},
    "VpcLinkId": "abcd123",
    "VpcLinkStatus": "AVAILABLE",
    "VpcLinkStatusMessage": "VPC link is ready to route traffic",
    "VpcLinkVersion": "V2"
}
```

Pour plus d'informations, consultez la section Utilisation de liens VPC pour HTTP APIs dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateVpcLinkà la section Référence des AWS CLI commandes.

Exemples d'API de gestion d'API Gateway utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l'API AWS Command Line Interface with API Gateway Management.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-connection`

L'exemple de code suivant montre comment utiliser`delete-connection`.

AWS CLI

Pour supprimer une WebSocket connexion

L'`delete-connection`exemple suivant déconnecte un client de l' WebSocket API spécifiée.

```
aws apigatewaymanagementapi delete-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les commandes @connections dans votre service principal dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteConnectionà la section Référence des AWS CLI commandes.

`get-connection`

L'exemple de code suivant montre comment utiliser`get-connection`.

AWS CLI

Pour obtenir des informations sur une WebSocket connexion

L'`get-connection`exemple suivant décrit une connexion à l' WebSocket API spécifiée.

```
aws apigatewaymanagementapi get-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Sortie :

```
{
    "ConnectedAt": "2020-04-30T20:10:33.236Z",
    "Identity": {
        "SourceIp": "192.0.2.1"
    },
    "LastActiveAt": "2020-04-30T20:10:42.997Z"
}
```

Pour plus d'informations, consultez la section Utiliser les commandes @connections dans votre service principal dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetConnectionà la section Référence des AWS CLI commandes.

`post-to-connection`

L'exemple de code suivant montre comment utiliser`post-to-connection`.

AWS CLI

Pour envoyer des données à une WebSocket connexion

L'`post-to-connection`exemple suivant envoie un message à un client connecté à l' WebSocket API spécifiée.

```
aws apigatewaymanagementapi post-to-connection \
```

```
    --connection-id L0SM9cOFvHcCIhw= \
    --data "Hello from API Gateway!" \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les commandes @connections dans votre service principal dans le manuel Amazon API Gateway Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PostToConnectionà la section Référence des AWS CLI commandes.

Exemples d'App Mesh utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de l'App Mesh.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-mesh`

L'exemple de code suivant montre comment utiliser`create-mesh`.

AWS CLI

Exemple 1 : pour créer un nouveau maillage de services

L'`create-mesh`exemple suivant crée un maillage de services.

```
aws appmesh create-mesh \
```

```
    --mesh-name app1
```

Sortie :

```
{
    "mesh":{
        "meshName":"app1",
        "metadata":{
            "arn":"arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt":1563809909.282,
            "lastUpdatedAt":1563809909.282,
            "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version":1
        },
        "spec":{},
        "status":{
            "status":"ACTIVE"
        }
    }
}
```

Exemple 2 : pour créer un nouveau maillage de services avec plusieurs balises

L'`create-mesh`exemple suivant crée un maillage de service avec plusieurs balises.

```
aws appmesh create-mesh \
    --mesh-name app2 \
    --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Sortie :

```
{
    "mesh":{
        "meshName":"app2",
        "metadata":{
            "arn":"arn:aws:appmesh:us-east-1:123456789012:mesh/app2",
            "createdAt":1563822121.877,
            "lastUpdatedAt":1563822121.877,
            "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version":1
        },
        "spec":{},
```

```
        "status":{
            "status":"ACTIVE"
        }
    }
}
```

Pour plus d'informations, consultez Service Meshes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous CreateMeshà la section Référence des AWS CLI commandes.

`create-route`

L'exemple de code suivant montre comment utiliser`create-route`.

AWS CLI

Pour créer une nouvelle route gRPC

L'`create-route`exemple suivant utilise un fichier d'entrée JSON pour créer une route gRPC. Le trafic GRPC dont les métadonnées commencent par 123 est acheminé vers un nœud virtuel nommé ServiceBGRPC. S'il y a des défaillances spécifiques de gRPC, HTTP ou TCP lors de la tentative de communication avec la cible de la route, la route est réessayée trois fois. Il y a un délai de 15 secondes entre chaque nouvelle tentative.

```
aws appmesh create-route \
    --cli-input-json file://create-route-grpc.json
```

Contenu de `create-route-grpc.json` :

```
{
    "meshName" : "apps",
    "routeName" : "grpcRoute",
    "spec" : {
       "grpcRoute" : {
          "action" : {
             "weightedTargets" : [
                {
                   "virtualNode" : "serviceBgrpc",
                   "weight" : 100
```

```
                }
             ]
          },
          "match" : {
             "metadata" : [
                {
                   "invert" : false,
                   "match" : {
                      "prefix" : "123"
                   },
                   "name" : "myMetadata"
                }
             ],
             "methodName" : "GetColor",
             "serviceName" : "com.amazonaws.services.ColorService"
          },
          "retryPolicy" : {
             "grpcRetryEvents" : [ "deadline-exceeded" ],
             "httpRetryEvents" : [ "server-error", "gateway-error" ],
             "maxRetries" : 3,
             "perRetryTimeout" : {
                "unit" : "s",
                "value" : 15
             },
             "tcpRetryEvents" : [ "connection-error" ]
          }
       },
       "priority" : 100
    },
    "virtualRouterName" : "serviceBgrpc"
}
```

Sortie :

```
{
    "route": {
        "meshName": "apps",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBgrpc/route/grpcRoute",
            "createdAt": 1572010806.008,
            "lastUpdatedAt": 1572010806.008,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
```

```
            "version": 1
        },
        "routeName": "grpcRoute",
        "spec": {
            "grpcRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "serviceBgrpc",
                            "weight": 100
                        }
                    ]
                },
                "match": {
                    "metadata": [
                        {
                            "invert": false,
                            "match": {
                                "prefix": "123"
                            },
                            "name": "mymetadata"
                        }
                    ],
                    "methodName": "GetColor",
                    "serviceName": "com.amazonaws.services.ColorService"
                },
                "retryPolicy": {
                    "grpcRetryEvents": [
                        "deadline-exceeded"
                    ],
                    "httpRetryEvents": [
                        "server-error",
                        "gateway-error"
                    ],
                    "maxRetries": 3,
                    "perRetryTimeout": {
                        "unit": "s",
                        "value": 15
                    },
                    "tcpRetryEvents": [
                        "connection-error"
                    ]
                }
            },
```

```
            "priority": 100
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "serviceBgrpc"
    }
}
```

Pour créer une nouvelle route HTTP ou HTTP/2

L'`create-route`exemple suivant utilise un fichier d'entrée JSON pour créer une route HTTP/2. Pour créer une route HTTP, remplacez Http2Route par HttpRoute selon les spécifications. Tout le trafic HTTP/2 adressé à un préfixe d'URL dont la valeur d'en-tête commence par 123 est acheminé vers un nœud virtuel nommé ServiceBHTTP2. En cas d'échec HTTP ou TCP spécifique lors de la tentative de communication avec la cible de la route, la route est réessayée trois fois. Il y a un délai de 15 secondes entre chaque nouvelle tentative.

```
aws appmesh create-route \
    --cli-input-json file://create-route-http2.json
```

Contenu de `create-route-http2.json` :

```
{
    "meshName": "apps",
    "routeName": "http2Route",
    "spec": {
        "http2Route": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "serviceBhttp2",
                        "weight": 100
                    }
                ]
            },
            "match": {
                "headers": [
                    {
                        "invert": false,
                        "match": {
                            "prefix": "123"
```

```
                        },
                        "name": "clientRequestId"
                    }
                ],
                "method": "POST",
                "prefix": "/",
                "scheme": "http"
            },
            "retryPolicy": {
                "httpRetryEvents": [
                    "server-error",
                    "gateway-error"
                ],
                "maxRetries": 3,
                "perRetryTimeout": {
                    "unit": "s",
                    "value": 15
                },
                "tcpRetryEvents": [
                    "connection-error"
                ]
            }
        },
        "priority": 200
    },
    "virtualRouterName": "serviceBhttp2"
}
```

Sortie :

```
{
    "route": {
        "meshName": "apps",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBhttp2/route/http2Route",
            "createdAt": 1572011008.352,
            "lastUpdatedAt": 1572011008.352,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "http2Route",
        "spec": {
```

```
            "http2Route": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "serviceBhttp2",
                            "weight": 100
                        }
                    ]
                },
                "match": {
                    "headers": [
                        {
                            "invert": false,
                            "match": {
                                "prefix": "123"
                            },
                            "name": "clientRequestId"
                        }
                    ],
                    "method": "POST",
                    "prefix": "/",
                    "scheme": "http"
                },
                "retryPolicy": {
                    "httpRetryEvents": [
                        "server-error",
                        "gateway-error"
                    ],
                    "maxRetries": 3,
                    "perRetryTimeout": {
                        "unit": "s",
                        "value": 15
                    },
                    "tcpRetryEvents": [
                        "connection-error"
                    ]
                }
            },
            "priority": 200
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "serviceBhttp2"
```

```
    }
}
```

Pour créer une nouvelle route TCP

L'`create-route`exemple suivant utilise un fichier d'entrée JSON pour créer une route TCP. 75 % du trafic est acheminé vers un nœud virtuel nommé ServiceBTCP, et 25 % du trafic est acheminé vers un nœud virtuel nommé ServiceBv2TCP. La spécification de différentes pondérations pour différentes cibles est un moyen efficace de déployer une nouvelle version d'une application. Vous pouvez ajuster les pondérations de manière à ce que 100 % du trafic soit finalement acheminé vers une cible dotée de la nouvelle version d'une application.

```
aws appmesh create-route \
    --cli-input-json file://create-route-tcp.json
```

Contenu du create-route-tcp fichier .json :

```
{
    "meshName": "apps",
    "routeName": "tcpRoute",
    "spec": {
        "priority": 300,
        "tcpRoute": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "serviceBtcp",
                        "weight": 75
                    },
                    {
                        "virtualNode": "serviceBv2tcp",
                        "weight": 25
                    }
                ]
            }
        }
    },
    "virtualRouterName": "serviceBtcp"
}
```

Sortie :

```
{
    "route": {
        "meshName": "apps",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBtcp/route/tcpRoute",
            "createdAt": 1572011436.26,
            "lastUpdatedAt": 1572011436.26,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "tcpRoute",
        "spec": {
            "priority": 300,
            "tcpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "serviceBtcp",
                            "weight": 75
                        },
                        {
                            "virtualNode": "serviceBv2tcp",
                            "weight": 25
                        }
                    ]
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "serviceBtcp"
    }
}
```

Pour plus d'informations, consultez Routes dans le guide de l'utilisateur de l'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous CreateRouteà la section Référence des AWS CLI commandes.

`create-virtual-gateway`

L'exemple de code suivant montre comment utiliser`create-virtual-gateway`.

AWS CLI

Pour créer une nouvelle passerelle virtuelle

L'`create-virtual-gateway`exemple suivant utilise un fichier d'entrée JSON pour créer une passerelle virtuelle avec un écouteur pour HTTP utilisant le port 9080.

```
aws appmesh create-virtual-gateway \
    --mesh-name meshName \
    --virtual-gateway-name virtualGatewayName \
    --cli-input-json file://create-virtual-gateway.json
```

Contenu de `create-virtual-gateway.json` :

```
{
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 9080,
            "protocol": "http"
          }
        }
      ]
    }
}
```

Sortie :

```
{
    "virtualGateway": {
        "meshName": "meshName",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/meshName/
virtualGateway/virtualGatewayName",
            "createdAt": "2022-04-06T10:42:42.015000-05:00",
            "lastUpdatedAt": "2022-04-06T10:42:42.015000-05:00",
            "meshOwner": "123456789012",
```

```
            "resourceOwner": "123456789012",
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 9080,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualGatewayName": "virtualGatewayName"
    }
}
```

Pour plus d'informations, consultez Virtual Gateways dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous CreateVirtualGatewayà la section Référence des AWS CLI commandes.

`create-virtual-node`

L'exemple de code suivant montre comment utiliser`create-virtual-node`.

AWS CLI

Exemple 1 : pour créer un nouveau nœud virtuel qui utilise le DNS pour la découverte

L'`create-virtual-node`exemple suivant utilise un fichier d'entrée JSON pour créer un nœud virtuel qui utilise le DNS pour la découverte de services.

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-dns.json
```

Contenu de `create-virtual-node-dns.json` :

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "dns": {
                "hostname": "serviceBv1.svc.cluster.local"
            }
        }
    },
    "virtualNodeName": "vnServiceBv1"
}
```

Sortie :

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "createdAt": 1563810019.874,
            "lastUpdatedAt": 1563810019.874,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
```

```
                "dns": {
                    "hostname": "serviceBv1.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceBv1"
    }
}
```

Exemple 2 : pour créer un nouveau nœud virtuel qui utilise AWS Cloud Map pour la découverte

L'`create-virtual-node`exemple suivant utilise un fichier d'entrée JSON pour créer un nœud virtuel qui utilise AWS Cloud Map pour la découverte de services.

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-cloud-map.json
```

Contenu de `create-virtual-node-cloud-map.json` :

```
{
    "meshName": "app1",
    "spec": {
        "backends": [
            {
                "virtualService": {
                    "virtualServiceName": "serviceA.svc.cluster.local"
                }
            }
        ],
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "awsCloudMap": {
                "attributes": [
```

```
                    {
                        "key": "Environment",
                        "value": "Testing"
                    }
                ],
                "namespaceName": "namespace1",
                "serviceName": "serviceA"
            }
        }
    },
    "virtualNodeName": "vnServiceA"
}
```

Sortie :

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceA",
            "createdAt": 1563810859.465,
            "lastUpdatedAt": 1563810859.465,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "backends": [
                {
                    "virtualService": {
                        "virtualServiceName": "serviceA.svc.cluster.local"
                    }
                }
            ],
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
```

```
                "awsCloudMap": {
                    "attributes": [
                        {
                            "key": "Environment",
                            "value": "Testing"
                        }
                    ],
                    "namespaceName": "namespace1",
                    "serviceName": "serviceA"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceA"
    }
}
```

Pour plus d'informations, consultez Virtual Nodes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous CreateVirtualNodeà la section Référence des AWS CLI commandes.

`create-virtual-router`

L'exemple de code suivant montre comment utiliser`create-virtual-router`.

AWS CLI

Pour créer un nouveau routeur virtuel

L'`create-virtual-router`exemple suivant utilise un fichier d'entrée JSON pour créer un routeur virtuel avec un écouteur pour HTTP via le port 80.

```
aws appmesh create-virtual-router \
    --cli-input-json file://create-virtual-router.json
```

Contenu de `create-virtual-router.json` :

```
{
```

```
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ]
    },
    "virtualRouterName": "vrServiceB"
}
```

Sortie :

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563810546.59,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

Pour plus d'informations, consultez la section Routeurs virtuels dans le guide de l'utilisateur de AWS l'App Mesh.

- Pour plus de détails sur l'API, reportez-vous CreateVirtualRouterà la section Référence des AWS CLI commandes.

`create-virtual-service`

L'exemple de code suivant montre comment utiliser`create-virtual-service`.

AWS CLI

Exemple 1 : pour créer un nouveau service virtuel avec un fournisseur de nœuds virtuels

L'`create-virtual-service`exemple suivant utilise un fichier d'entrée JSON pour créer un service virtuel avec un fournisseur de nœuds virtuels.

```
aws appmesh create-virtual-service \
    --cli-input-json file://create-virtual-service-virtual-node.json
```

Contenu de `create-virtual-service-virtual-node.json` :

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualNode": {
                "virtualNodeName": "vnServiceA"
            }
        }
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}
```

Sortie :

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
```

```
            "createdAt": 1563810859.474,
            "lastUpdatedAt": 1563810967.179,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "provider": {
                "virtualNode": {
                    "virtualNodeName": "vnServiceA"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceA.svc.cluster.local"
    }
}
```

Pour plus d'informations, consultez Virtual Node dans le guide de l'utilisateur d'AWS App Mesh.

Exemple 2 : pour créer un nouveau service virtuel auprès d'un fournisseur de routeur virtuel

L'`create-virtual-service`exemple suivant utilise un fichier d'entrée JSON pour créer un service virtuel avec un fournisseur de routeur virtuel.

```
aws appmesh create-virtual-service \
    --cli-input-json file://create-virtual-service-virtual-router.json
```

Contenu de `create-virtual-service-virtual-router.json` :

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualRouter": {
                "virtualRouterName": "vrServiceB"
            }
        }
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
}
```

Sortie :

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "createdAt": 1563908363.999,
            "lastUpdatedAt": 1563908363.999,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "provider": {
                "virtualRouter": {
                    "virtualRouterName": "vrServiceB"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceB.svc.cluster.local"
    }
}
```

Pour plus d'informations, consultez Virtual Services< https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual _services.html> dans le guide de l'utilisateur d'App Mesh AWS

- Pour plus de détails sur l'API, reportez-vous CreateVirtualServiceà la section Référence des AWS CLI commandes.

`delete-mesh`

L'exemple de code suivant montre comment utiliser`delete-mesh`.

AWS CLI

Pour supprimer un maillage de services

L'`delete-mesh`exemple suivant supprime le maillage de service spécifié.

```
aws appmesh delete-mesh \
    --mesh-name app1
```

Sortie :

```
{
    "mesh": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt": 1563809909.282,
            "lastUpdatedAt": 1563824981.248,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "egressFilter": {
                "type": "ALLOW_ALL"
            }
        },
        "status": {
            "status": "DELETED"
        }
    }
}
```

Pour plus d'informations, consultez Service Meshes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DeleteMeshà la section Référence des AWS CLI commandes.

`delete-route`

L'exemple de code suivant montre comment utiliser`delete-route`.

AWS CLI

Pour supprimer un itinéraire

L'`delete-route`exemple suivant supprime l'itinéraire spécifié.

```
aws appmesh delete-route \
    --mesh-name app1 \
    --virtual-router-name vrServiceB \
    --route-name toVnServiceB-weighted
```

Sortie :

```
{
    "route": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
            "createdAt": 1563811384.015,
            "lastUpdatedAt": 1563823915.936,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "routeName": "toVnServiceB-weighted",
        "spec": {
            "httpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "vnServiceBv1",
                            "weight": 80
                        },
                        {
                            "virtualNode": "vnServiceBv2",
                            "weight": 20
                        }
                    ]
                },
                "match": {
                    "prefix": "/"
                }
            }
        },
        "status": {
            "status": "DELETED"
        },
        "virtualRouterName": "vrServiceB"
    }
```

```
}
```

Pour plus d'informations, consultez Routes dans le guide de l'utilisateur de l'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DeleteRouteà la section Référence des AWS CLI commandes.

`delete-virtual-node`

L'exemple de code suivant montre comment utiliser`delete-virtual-node`.

AWS CLI

Pour supprimer un nœud virtuel

L'`delete-virtual-node`exemple suivant supprime le nœud virtuel spécifié.

```
aws appmesh delete-virtual-node \
    --mesh-name app1 \
    --virtual-node-name vnServiceBv2
```

Sortie :

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv2",
            "createdAt": 1563810117.297,
            "lastUpdatedAt": 1563824700.678,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "backends": [],
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
```

```
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv2.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "DELETED"
        },
        "virtualNodeName": "vnServiceBv2"
    }
}
```

Pour plus d'informations, consultez Virtual Nodes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DeleteVirtualNodeà la section Référence des AWS CLI commandes.

`delete-virtual-router`

L'exemple de code suivant montre comment utiliser`delete-virtual-router`.

AWS CLI

Pour supprimer un routeur virtuel

L'`delete-virtual-router`exemple suivant supprime le routeur virtuel spécifié.

```
aws appmesh delete-virtual-router \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

Sortie :

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563824253.467,
```

```
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "DELETED"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

Pour plus d'informations, consultez la section Routeurs virtuels dans le guide de l'utilisateur de AWS l'App Mesh.

- Pour plus de détails sur l'API, reportez-vous DeleteVirtualRouterà la section Référence des AWS CLI commandes.

`delete-virtual-service`

L'exemple de code suivant montre comment utiliser`delete-virtual-service`.

AWS CLI

Pour supprimer un service virtuel

L'`delete-virtual-service`exemple suivant supprime le service virtuel spécifié.

```
aws appmesh delete-virtual-service \
    --mesh-name app1 \
    --virtual-service-name serviceB.svc.cluster.local
```

Sortie :

```
{
```

```
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "createdAt": 1563908363.999,
            "lastUpdatedAt": 1563913940.866,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "spec": {},
        "status": {
            "status": "DELETED"
        },
        "virtualServiceName": "serviceB.svc.cluster.local"
    }
}
```

Pour plus d'informations, consultez Virtual Service dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DeleteVirtualServiceà la section Référence des AWS CLI commandes.

`describe-mesh`

L'exemple de code suivant montre comment utiliser`describe-mesh`.

AWS CLI

Pour décrire un maillage de services

L'`describe-mesh`exemple suivant renvoie des informations sur le maillage de service spécifié.

```
aws appmesh describe-mesh \
    --mesh-name app1
```

Sortie :

```
{
    "mesh": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
```

```
            "createdAt": 1563809909.282,
            "lastUpdatedAt": 1563809909.282,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {},
        "status": {
            "status": "ACTIVE"
        }
    }
}
```

Pour plus d'informations, consultez Service Meshes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DescribeMeshà la section Référence des AWS CLI commandes.

`describe-route`

L'exemple de code suivant montre comment utiliser`describe-route`.

AWS CLI

Pour décrire un itinéraire

L'`describe-route`exemple suivant renvoie les détails de l'itinéraire spécifié.

```
aws appmesh describe-route \
    --mesh-name app1 \
    --virtual-router-name vrServiceB \
    --route-name toVnServiceB-weighted
```

Sortie :

```
{
    "route": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
            "createdAt": 1563811384.015,
```

```
            "lastUpdatedAt": 1563811384.015,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "toVnServiceB-weighted",
        "spec": {
            "httpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "vnServiceBv1",
                            "weight": 90
                        },
                        {
                            "virtualNode": "vnServiceBv2",
                            "weight": 10
                        }
                    ]
                },
                "match": {
                    "prefix": "/"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

Pour plus d'informations, consultez Routes dans le guide de l'utilisateur de l'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DescribeRouteà la section Référence des AWS CLI commandes.

`describe-virtual-node`

L'exemple de code suivant montre comment utiliser`describe-virtual-node`.

AWS CLI

Pour décrire un nœud virtuel

L'`describe-virtual-node`exemple suivant renvoie des informations sur le nœud virtuel spécifié.

```
aws appmesh describe-virtual-node \
    --mesh-name app1 \
    --virtual-node-name vnServiceBv1
```

Sortie :

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "createdAt": 1563810019.874,
            "lastUpdatedAt": 1563810019.874,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "backends": [],
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv1.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceBv1"
    }
}
```

Pour plus d'informations, consultez Virtual Nodes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DescribeVirtualNodeà la section Référence des AWS CLI commandes.

`describe-virtual-router`

L'exemple de code suivant montre comment utiliser`describe-virtual-router`.

AWS CLI

Pour décrire un routeur virtuel

L'`describe-virtual-router`exemple suivant renvoie des informations sur le routeur virtuel spécifié.

```
aws appmesh describe-virtual-router \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

Sortie :

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563810546.59,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ]
        },
```

```
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

Pour plus d'informations, consultez la section Routeurs virtuels dans le guide de l'utilisateur de AWS l'App Mesh.

- Pour plus de détails sur l'API, reportez-vous DescribeVirtualRouterà la section Référence des AWS CLI commandes.

`describe-virtual-service`

L'exemple de code suivant montre comment utiliser`describe-virtual-service`.

AWS CLI

Pour décrire un service virtuel

L'`describe-virtual-service`exemple suivant renvoie des informations sur le service virtuel spécifié.

```
aws appmesh describe-virtual-service \
    --mesh-name app1 \
    --virtual-service-name serviceB.svc.cluster.local
```

Sortie :

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "createdAt": 1563908363.999,
            "lastUpdatedAt": 1563908363.999,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
```

```
            "provider": {
                "virtualRouter": {
                    "virtualRouterName": "vrServiceB"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceB.svc.cluster.local"
    }
}
```

Pour plus d'informations, consultez Virtual Services dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous DescribeVirtualServiceà la section Référence des AWS CLI commandes.

`list-meshes`

L'exemple de code suivant montre comment utiliser`list-meshes`.

AWS CLI

Pour répertorier les maillages de service

L'`list-meshes`exemple suivant répertorie tous les maillages de service de la AWS région actuelle.

```
aws appmesh list-meshes
```

Sortie :

```
{
    "meshes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "meshName": "app1"
        }
    ]
}
```

Pour plus d'informations, consultez Service Meshes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous ListMeshesà la section Référence des AWS CLI commandes.

`list-routes`

L'exemple de code suivant montre comment utiliser`list-routes`.

AWS CLI

Pour répertorier les itinéraires

L'`list-routes`exemple suivant répertorie toutes les routes pour le routeur virtuel spécifié.

```
aws appmesh list-routes \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

Sortie :

```
{
    "routes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB",
            "meshName": "app1",
            "routeName": "toVnServiceB-weighted",
            "virtualRouterName": "vrServiceB"
        }
    ]
}
```

Pour plus d'informations, consultez Routes dans le guide de l'utilisateur de l'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous ListRoutesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises attribuées à la ressource spécifiée.

```
aws appmesh list-tags-for-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

Sortie :

```
{
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        },
        {
            "key": "key3",
            "value": "value3"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-virtual-nodes`

L'exemple de code suivant montre comment utiliser`list-virtual-nodes`.

AWS CLI

Pour répertorier les nœuds virtuels

L'`list-virtual-nodes`exemple suivant répertorie tous les nœuds virtuels du maillage de service spécifié.

```
aws appmesh list-virtual-nodes \
    --mesh-name app1
```

Sortie :

```
{
    "virtualNodes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "meshName": "app1",
            "virtualNodeName": "vnServiceBv1"
        },
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv2",
            "meshName": "app1",
            "virtualNodeName": "vnServiceBv2"
        }
    ]
}
```

Pour plus d'informations, consultez Virtual Nodes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous ListVirtualNodesà la section Référence des AWS CLI commandes.

`list-virtual-routers`

L'exemple de code suivant montre comment utiliser`list-virtual-routers`.

AWS CLI

Pour répertorier les routeurs virtuels

L'`list-virtual-routers`exemple suivant répertorie tous les routeurs virtuels du maillage de service spécifié.

```
aws appmesh list-virtual-routers \
    --mesh-name app1
```

Sortie :

```
{
    "virtualRouters": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "meshName": "app1",
            "virtualRouterName": "vrServiceB"
        }
    ]
}
```

Pour plus d'informations, consultez la section Routeurs virtuels dans le guide de l'utilisateur de AWS l'App Mesh.

- Pour plus de détails sur l'API, reportez-vous ListVirtualRoutersà la section Référence des AWS CLI commandes.

`list-virtual-services`

L'exemple de code suivant montre comment utiliser`list-virtual-services`.

AWS CLI

Pour répertorier les services virtuels

L'`list-virtual-services`exemple suivant répertorie tous les services virtuels du maillage de services spécifié.

```
aws appmesh list-virtual-services \
    --mesh-name app1
```

Sortie :

```
{
    "virtualServices": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
            "meshName": "app1",
            "virtualServiceName": "serviceA.svc.cluster.local"
        },
```

```
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "meshName": "app1",
            "virtualServiceName": "serviceB.svc.cluster.local"
        }
    ]
}
```

Pour plus d'informations, consultez Virtual Services dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous ListVirtualServicesà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute la balise `key1` avec la valeur `value1` à la ressource spécifiée.

```
aws appmesh tag-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
    --tags key=key1,value=value1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour annuler le balisage d'une ressource

L'`untag-resource`exemple suivant supprime une balise contenant la clé `key1` de la ressource spécifiée.

```
aws appmesh untag-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
    --tag-keys key1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-mesh`

L'exemple de code suivant montre comment utiliser`update-mesh`.

AWS CLI

Pour mettre à jour un maillage de services

L'`update-mesh`exemple suivant utilise un fichier d'entrée JSON pour mettre à jour un maillage de service afin de permettre à tout le trafic sortant externe d'être transféré sans modification via le proxy Envoy.

```
aws appmesh update-mesh \
    --cli-input-json file://update-mesh.json
```

Contenu de `update-mesh.json` :

```
{
    "meshName": "app1",
    "spec": {
        "egressFilter": {
            "type": "ALLOW_ALL"
        }
    }
}
```

Sortie :

```
{
    "mesh": {
```

```
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt": 1563809909.282,
            "lastUpdatedAt": 1563812829.687,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "egressFilter": {
                "type": "ALLOW_ALL"
            }
        },
        "status": {
            "status": "ACTIVE"
        }
    }
}
```

Pour plus d'informations, consultez Service Meshes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous UpdateMeshà la section Référence des AWS CLI commandes.

`update-route`

L'exemple de code suivant montre comment utiliser`update-route`.

AWS CLI

Pour mettre à jour un itinéraire

L'`update-route`exemple suivant utilise un fichier d'entrée JSON pour mettre à jour les pondérations d'un itinéraire.

```
aws appmesh update-route \
    --cli-input-json file://update-route-weighted.json
```

Contenu de `update-route-weighted.json` :

```
{
```

```
    "meshName": "app1",
    "routeName": "toVnServiceB-weighted",
    "spec": {
        "httpRoute": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "vnServiceBv1",
                        "weight": 80
                    },
                    {
                        "virtualNode": "vnServiceBv2",
                        "weight": 20
                    }
                ]
            },
            "match": {
                "prefix": "/"
            }
        }
    },
    "virtualRouterName": "vrServiceB"
}
```

Sortie :

```
{
    "route": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
            "createdAt": 1563811384.015,
            "lastUpdatedAt": 1563819600.022,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "routeName": "toVnServiceB-weighted",
        "spec": {
            "httpRoute": {
                "action": {
                    "weightedTargets": [
                        {
```

```
                            "virtualNode": "vnServiceBv1",
                            "weight": 80
                        },
                        {
                            "virtualNode": "vnServiceBv2",
                            "weight": 20
                        }
                    ]
                },
                "match": {
                    "prefix": "/"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

Pour plus d'informations, consultez Routes dans le guide de l'utilisateur de l'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous UpdateRouteà la section Référence des AWS CLI commandes.

`update-virtual-node`

L'exemple de code suivant montre comment utiliser`update-virtual-node`.

AWS CLI

Pour mettre à jour un nœud virtuel

L'`update-virtual-node`exemple suivant utilise un fichier d'entrée JSON pour ajouter un contrôle de santé à un nœud virtuel.

```
aws appmesh update-virtual-node \
    --cli-input-json file://update-virtual-node.json
```

Contenu de `update-virtual-node.json` :

```
{
```

```
    "clientToken": "500",
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "healthCheck": {
                    "healthyThreshold": 5,
                    "intervalMillis": 10000,
                    "path": "/",
                    "port": 80,
                    "protocol": "http",
                    "timeoutMillis": 3000,
                    "unhealthyThreshold": 3
                },
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "dns": {
                "hostname": "serviceBv1.svc.cluster.local"
            }
        }
    },
    "virtualNodeName": "vnServiceBv1"
}
```

Sortie :

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "createdAt": 1563810019.874,
            "lastUpdatedAt": 1563819234.825,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
```

```
            "listeners": [
                {
                    "healthCheck": {
                        "healthyThreshold": 5,
                        "intervalMillis": 10000,
                        "path": "/",
                        "port": 80,
                        "protocol": "http",
                        "timeoutMillis": 3000,
                        "unhealthyThreshold": 3
                    },
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv1.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceBv1"
    }
}
```

Pour plus d'informations, consultez Virtual Nodes dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous UpdateVirtualNodeà la section Référence des AWS CLI commandes.

`update-virtual-router`

L'exemple de code suivant montre comment utiliser`update-virtual-router`.

AWS CLI

Pour mettre à jour un routeur virtuel

L'`update-virtual-router`exemple suivant utilise un fichier d'entrée JSON pour mettre à jour le port d'écoute d'un routeur virtuel.

```
aws appmesh update-virtual-router \
    --cli-input-json file://update-virtual-router.json
```

Contenu de `update-virtual-router.json` :

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 8080,
                    "protocol": "http"
                }
            }
        ]
    },
    "virtualRouterName": "vrServiceB"
}
```

Sortie :

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563819431.352,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 8080,
                        "protocol": "http"
```

```
                    }
                }
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

Pour plus d'informations, consultez la section Routeurs virtuels dans le guide de l'utilisateur de AWS l'App Mesh.

- Pour plus de détails sur l'API, reportez-vous UpdateVirtualRouterà la section Référence des AWS CLI commandes.

`update-virtual-service`

L'exemple de code suivant montre comment utiliser`update-virtual-service`.

AWS CLI

Pour mettre à jour un service virtuel

L'`update-virtual-service`exemple suivant utilise un fichier d'entrée JSON pour mettre à jour un service virtuel afin d'utiliser un fournisseur de routeur virtuel.

```
aws appmesh update-virtual-service \
    --cli-input-json file://update-virtual-service.json
```

Contenu de `update-virtual-service.json` :

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualRouter": {
                "virtualRouterName": "vrServiceA"
            }
        }
```

```
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}
```

Sortie :

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
            "createdAt": 1563810859.474,
            "lastUpdatedAt": 1563820257.411,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "spec": {
            "provider": {
                "virtualRouter": {
                    "virtualRouterName": "vrServiceA"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceA.svc.cluster.local"
    }
}
```

Pour plus d'informations, consultez Virtual Services dans le guide de l'utilisateur d'AWS App Mesh.

- Pour plus de détails sur l'API, reportez-vous UpdateVirtualServiceà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'App Runner AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'App Runner.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-custom-domain`

L'exemple de code suivant montre comment utiliser`associate-custom-domain`.

AWS CLI

Pour associer un nom de domaine et le sous-domaine www à un service

L'`associate-custom-domain`exemple suivant associe un nom de domaine personnalisé que vous contrôlez à un service App Runner. Le nom de domaine est le domaine racine`example.com`, y compris le sous-domaine spécifique. `www.example.com`

```
aws apprunner associate-custom-domain \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com",
    "EnableWWWSubdomain": true
}
```

Sortie :

```
{
```

```
    "CustomDomain": {
        "CertificateValidationRecords": [
            {
                "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
            },
            {
                "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
            }
        ],
        "DomainName": "example.com",
        "EnableWWWSubdomain": true,
        "Status": "CREATING"
    },
    "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateCustomDomainà la section Référence des AWS CLI commandes.

`create-auto-scaling-configuration`

L'exemple de code suivant montre comment utiliser`create-auto-scaling-configuration`.

AWS CLI

Pour créer une configuration de mise à l'échelle automatique à haute disponibilité

L'`create-auto-scaling-configuration`exemple suivant crée une configuration de dimensionnement automatique optimisée pour une haute disponibilité en définissant `MinSize` la valeur 5. Avec cette configuration, App Runner tente de répartir vos instances de service sur le plus grand nombre possible de zones de disponibilité, jusqu'à cinq, selon la AWS région.

L'appel renvoie un `AutoScalingConfiguration` objet dont les autres paramètres sont définis par défaut. Dans l'exemple, il s'agit du premier appel pour créer une configuration nomméehigh-availability. La révision est définie sur 1, et il s'agit de la dernière révision.

```
aws apprunner create-auto-scaling-configuration \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "AutoScalingConfigurationName": "high-availability",
    "MinSize": 5
}
```

Sortie :

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 1,
        "CreatedAt": "2020-11-03T00:29:17Z",
        "Latest": true,
        "Status": "ACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
        "MinSize": 5
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAutoScalingConfigurationà la section Référence des AWS CLI commandes.

`create-connection`

L'exemple de code suivant montre comment utiliser`create-connection`.

AWS CLI

Pour créer une GitHub connexion

L'`create-connection`exemple suivant crée une connexion à un référentiel de GitHub code privé. L'état de la connexion après un appel réussi est`PENDING_HANDSHAKE`. Cela est dû au fait qu'aucune poignée de main d'authentification avec le fournisseur n'a toujours eu lieu. Terminez la poignée de main à l'aide de la console App Runner.

```
aws apprunner create-connection \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ConnectionName": "my-github-connection",
    "ProviderType": "GITHUB"
}
```

Sortie :

```
{
    "Connection": {
        "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-
github-connection",
        "ConnectionName": "my-github-connection",
        "Status": "PENDING_HANDSHAKE",
        "CreatedAt": "2020-11-03T00:32:51Z",
        "ProviderType": "GITHUB"
    }
}
```

Pour plus d'informations, consultez la section Gestion des connexions App Runner dans le guide du développeur d'AWS App Runner.

- Pour plus de détails sur l'API, reportez-vous CreateConnectionà la section Référence des AWS CLI commandes.

`create-service`

L'exemple de code suivant montre comment utiliser`create-service`.

AWS CLI

Exemple 1 : pour créer un service de référentiel de code source

L'`create-service`exemple suivant crée un service App Runner basé sur un référentiel de code source Python.

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceName": "python-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection/e7656250f67242d7819feade6800f59e"
        },
        "AutoDeploymentsEnabled": true,
        "CodeRepository": {
            "RepositoryUrl": "https://github.com/my-account/python-hello",
            "SourceCodeVersion": {
                "Type": "BRANCH",
                "Value": "main"
            },
            "CodeConfiguration": {
                "ConfigurationSource": "API",
                "CodeConfigurationValues": {
                    "Runtime": "PYTHON_3",
                    "BuildCommand": "pip install -r requirements.txt",
                    "StartCommand": "python server.py",
                    "Port": "8080",
                    "RuntimeEnvironmentVariables": [
                        {
                            "NAME": "Jane"
                        }
                    ]
                }
            }
        }
    },
    "InstanceConfiguration": {
```

```
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
```

```
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

Exemple 2 : pour créer un service de référentiel de code source

L'`create-service`exemple suivant crée un service App Runner basé sur un référentiel de code source Python.

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceName": "python-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection/e7656250f67242d7819feade6800f59e"
        },
        "AutoDeploymentsEnabled": true,
        "CodeRepository": {
            "RepositoryUrl": "https://github.com/my-account/python-hello",
            "SourceCodeVersion": {
                "Type": "BRANCH",
                "Value": "main"
            },
            "CodeConfiguration": {
                "ConfigurationSource": "API",
                "CodeConfigurationValues": {
                    "Runtime": "PYTHON_3",
                    "BuildCommand": "pip install -r requirements.txt",
                    "StartCommand": "python server.py",
                    "Port": "8080",
                    "RuntimeEnvironmentVariables": [
```

```
                        {
                            "NAME": "Jane"
                        }
                    ]
                }
            }
        },
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
```

```
                    ],
                    "StartCommand": "python server.py"
                },
                "ConfigurationSource": "Api"
            },
            "RepositoryUrl": "https://github.com/my-account/python-hello",
            "SourceCodeVersion": {
                "Type": "BRANCH",
                "Value": "main"
            }
        }
    },
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
}
```

Exemple 3 : pour créer un service de référentiel d'images source

L'`create-service`exemple suivant crée un service App Runner basé sur une image stockée dans Elastic Container Registry (ECR).

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceName": "golang-container-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
        },
        "AutoDeploymentsEnabled": true,
        "ImageRepository": {
            "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/golang-
app:latest",
            "ImageConfiguration": {
                "Port": "8080",
                "RuntimeEnvironmentVariables": [
```

```
                    {
                        "NAME": "Jane"
                    }
                ]
            },
            "ImageRepositoryType": "ECR"
        }
    },
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-06T23:15:30Z",
        "UpdatedAt": "2020-11-06T23:15:30Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/51728f8a20ce46d39b25398a6c8e9d1a",
        "ServiceId": "51728f8a20ce46d39b25398a6c8e9d1a",
        "ServiceName": "golang-container-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
            },
            "AutoDeploymentsEnabled": true,
            "ImageRepository": {
                "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/
golang-app:latest",
                "ImageConfiguration": {
                    "Port": "8080",
                    "RuntimeEnvironmentVariables": [
                        {
                            "NAME": "Jane"
                        }
                    ]
                },
                "ImageRepositoryType": "ECR"
```

```
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateServiceà la section Référence des AWS CLI commandes.

`delete-auto-scaling-configuration`

L'exemple de code suivant montre comment utiliser`delete-auto-scaling-configuration`.

AWS CLI

Exemple 1 : pour supprimer la dernière révision active d'une configuration de mise à l'échelle automatique

L'`delete-auto-scaling-configuration`exemple suivant supprime la dernière révision active d'une configuration de mise à l'échelle automatique d'App Runner. Pour supprimer la dernière révision active, spécifiez un Amazon Resource Name (ARN) qui se termine par le nom de configuration, sans le composant de révision.

Dans l'exemple, deux révisions existent avant cette action. Par conséquent, la révision 2 (la plus récente) est supprimée. Cependant, elle apparaît maintenant`"Latest": false`, car, après avoir été supprimée, il ne s'agit plus de la dernière révision active.

```
aws apprunner delete-auto-scaling-configuration \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

Sortie :

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 2,
        "CreatedAt": "2021-02-25T17:42:59Z",
        "DeletedAt": "2021-03-02T08:07:06Z",
        "Latest": false,
        "Status": "INACTIVE",
        "MaxConcurrency": 30,
        "MaxSize": 90,
        "MinSize": 5
    }
}
```

Exemple 2 : pour supprimer une révision spécifique d'une configuration de dimensionnement automatique

L'`delete-auto-scaling-configuration`exemple suivant supprime une révision spécifique d'une configuration de mise à l'échelle automatique d'App Runner. Pour supprimer une révision spécifique, spécifiez un ARN qui inclut le numéro de révision.

Dans l'exemple, plusieurs révisions existent avant cette action. L'action supprime la révision1.

```
aws apprunner delete-auto-scaling-configuration \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Sortie :

```
{
    "AutoScalingConfiguration": {
```

```
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 1,
        "CreatedAt": "2020-11-03T00:29:17Z",
        "DeletedAt": "2021-03-02T08:07:06Z",
        "Latest": false,
        "Status": "INACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
        "MinSize": 5
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAutoScalingConfigurationà la section Référence des AWS CLI commandes.

`delete-connection`

L'exemple de code suivant montre comment utiliser`delete-connection`.

AWS CLI

Pour supprimer une connexion

L'`delete-connection`exemple suivant supprime une connexion App Runner. L'état de la connexion après un appel réussi est`DELETED`. Cela est dû au fait que la connexion n'est plus disponible.

```
aws apprunner delete-connection \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-
connection"
}
```

Sortie :

```
{
    "Connection": {
        "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-
github-connection",
        "ConnectionName": "my-github-connection",
        "Status": "DELETED",
        "CreatedAt": "2020-11-03T00:32:51Z",
        "ProviderType": "GITHUB"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteConnectionà la section Référence des AWS CLI commandes.

`delete-service`

L'exemple de code suivant montre comment utiliser`delete-service`.

AWS CLI

Pour supprimer un service

L'`delete-service`exemple suivant supprime un service App Runner.

```
aws apprunner delete-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
```

```
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteServiceà la section Référence des AWS CLI commandes.

`describe-auto-scaling-configuration`

L'exemple de code suivant montre comment utiliser`describe-auto-scaling-configuration`.

AWS CLI

Exemple 1 : pour décrire la dernière révision active d'une configuration de mise à l'échelle automatique

L'`describe-auto-scaling-configuration`exemple suivant fournit une description de la dernière révision active d'une configuration de mise à l'échelle automatique d'App Runner. Pour décrire la dernière révision active, spécifiez un ARN qui se termine par le nom de la configuration, sans le composant de révision.

Dans l'exemple, deux révisions existent. Par conséquent, la révision 2 (la plus récente) est décrite. L'objet obtenu s'affiche`"Latest": true`.

```
aws apprunner describe-auto-scaling-configuration \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

Sortie :

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 2,
        "CreatedAt": "2021-02-25T17:42:59Z",
        "Latest": true,
```

```
        "Status": "ACTIVE",
        "MaxConcurrency": 30,
        "MaxSize": 90,
        "MinSize": 5
    }
}
```

Exemple 2 : pour décrire une révision spécifique d'une configuration de mise à l'échelle automatique

L'`describe-auto-scaling-configuration`exemple suivant décrit une révision spécifique d'une configuration de mise à l'échelle automatique d'App Runner. Pour décrire une révision spécifique, spécifiez un ARN qui inclut le numéro de révision.

Dans l'exemple, plusieurs révisions existent et la révision 1 est demandée. L'objet obtenu s'affiche`"Latest": false`.

```
aws apprunner describe-auto-scaling-configuration \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Sortie :

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 1,
        "CreatedAt": "2020-11-03T00:29:17Z",
        "Latest": false,
        "Status": "ACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
        "MinSize": 5
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAutoScalingConfigurationà la section Référence des AWS CLI commandes.

`describe-custom-domains`

L'exemple de code suivant montre comment utiliser`describe-custom-domains`.

AWS CLI

Pour obtenir les descriptions des noms de domaine personnalisés associés à un service

L'`describe-custom-domains`exemple suivant permet d'obtenir les descriptions et le statut des noms de domaine personnalisés associés à un service App Runner.

```
aws apprunner describe-custom-domains \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com",
    "EnableWWWSubdomain": true
}
```

Sortie :

```
{
    "CustomDomains": [
        {
            "CertificateValidationRecords": [
                {
                    "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
                    "Status": "PENDING_VALIDATION",
                    "Type": "CNAME",
                    "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
```

```
                },
                {
                    "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
                    "Status": "PENDING_VALIDATION",
                    "Type": "CNAME",
                    "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
                }
            ],
            "DomainName": "example.com",
            "EnableWWWSubdomain": true,
            "Status": "PENDING_CERTIFICATE_DNS_VALIDATION"
        },
        {
            "CertificateValidationRecords": [
                {
                    "Name": "_a94f784c70d3f507c72dc28f55db2f6b.deals.example.com",
                    "Status": "SUCCESS",
                    "Type": "CNAME",
                    "Value": "_2db02504c1270c137383c6307b6834b0.bsgbmzkfwj.acm-
validations.aws."
                }
            ],
            "DomainName": "deals.example.com",
            "EnableWWWSubdomain": false,
            "Status": "ACTIVE"
        }
    ],
    "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeCustomDomainsà la section Référence des AWS CLI commandes.

`describe-service`

L'exemple de code suivant montre comment utiliser`describe-service`.

AWS CLI

Pour décrire un service

L'`describe-service`exemple suivant fournit une description d'un service App Runner.

```
aws apprunner describe-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
```

```
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "RUNNING",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeServiceà la section Référence des AWS CLI commandes.

`disassociate-custom-domain`

L'exemple de code suivant montre comment utiliser`disassociate-custom-domain`.

AWS CLI

Pour dissocier un nom de domaine d'un service

L'`disassociate-custom-domain`exemple suivant dissocie le domaine `example.com` d'un service App Runner. L'appel dissocie également le sous-domaine `www.example.com` associé au domaine racine.

```
aws apprunner disassociate-custom-domain \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
```

```
    "DomainName": "example.com"
}
```

Sortie :

```
{
    "CustomDomain": {
        "CertificateValidationRecords": [
            {
                "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
            },
            {
                "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
            }
        ],
        "DomainName": "example.com",
        "EnableWWWSubdomain": true,
        "Status": "DELETING"
    },
    "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateCustomDomainà la section Référence des AWS CLI commandes.

`list-auto-scaling-configurations`

L'exemple de code suivant montre comment utiliser`list-auto-scaling-configurations`.

AWS CLI

Pour obtenir une liste paginée des configurations de mise à l'échelle automatique d'App Runner

L'`list-auto-scaling-configurations`exemple suivant répertorie toutes les configurations de mise à l'échelle automatique d'App Runner de votre AWS compte. Jusqu'à cinq configurations de mise à l'échelle automatique sont répertoriées dans chaque réponse. `AutoScalingConfigurationName`et `LatestOnly` ne sont pas spécifiés. Leurs valeurs par défaut font en sorte que la dernière révision de toutes les configurations actives soit répertoriée.

Dans cet exemple, la réponse inclut deux résultats et il n'y en a pas d'autres. Aucun résultat n'`NextToken`est donc renvoyé.

```
aws apprunner list-auto-scaling-configurations \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "MaxResults": 5
}
```

Sortie :

```
{
    "AutoScalingConfigurationSummaryList": [
        {
            "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
            "AutoScalingConfigurationName": "high-availability",
            "AutoScalingConfigurationRevision": 2
        },
        {
            "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/low-
cost/1/50d7804e7656fead0f59672e62f2e819",
            "AutoScalingConfigurationName": "low-cost",
            "AutoScalingConfigurationRevision": 1
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAutoScalingConfigurationsà la section Référence des AWS CLI commandes.

`list-connections`

L'exemple de code suivant montre comment utiliser`list-connections`.

AWS CLI

Exemple 1 : pour répertorier toutes les connexions

L'`list-connections`exemple suivant répertorie toutes les connexions App Runner du AWS compte.

```
aws apprunner list-connections
```

Sortie :

```
{
    "ConnectionSummaryList": [
        {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection",
            "ConnectionName": "my-github-connection",
            "Status": "AVAILABLE",
            "CreatedAt": "2020-11-03T00:32:51Z",
            "ProviderType": "GITHUB"
        },
        {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
            "ConnectionName": "my-github-org-connection",
            "Status": "AVAILABLE",
            "CreatedAt": "2020-11-03T02:54:17Z",
            "ProviderType": "GITHUB"
        }
    ]
}
```

Exemple 2 : pour répertorier une connexion par nom

L'`list-connections`exemple suivant répertorie une connexion par son nom.

```
aws apprunner list-connections \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ConnectionName": "my-github-org-connection"
}
```

Sortie :

```
{
    "ConnectionSummaryList": [
        {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
            "ConnectionName": "my-github-org-connection",
            "Status": "AVAILABLE",
            "CreatedAt": "2020-11-03T02:54:17Z",
            "ProviderType": "GITHUB"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListConnectionsà la section Référence des AWS CLI commandes.

`list-operations`

L'exemple de code suivant montre comment utiliser`list-operations`.

AWS CLI

Pour répertorier les opérations effectuées sur un service

L'`list-operations`exemple suivant répertorie toutes les opérations effectuées jusqu'à présent sur un service App Runner. Dans cet exemple, le service est nouveau et une seule opération de type `CREATE_SERVICE` s'est produite.

```
aws apprunner list-operations \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "OperationSummaryList": [
        {
            "EndedAt": 1606156217,
            "Id": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
            "StartedAt": 1606156014,
            "Status": "SUCCEEDED",
            "TargetArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
            "Type": "CREATE_SERVICE",
            "UpdatedAt": 1606156217
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListOperationsà la section Référence des AWS CLI commandes.

`list-services`

L'exemple de code suivant montre comment utiliser`list-services`.

AWS CLI

Pour obtenir une liste paginée des services App Runner

L'`list-services`exemple suivant répertorie tous les services App Runner du AWS compte. Jusqu'à deux services sont répertoriés dans chaque réponse. Cet exemple montre la première demande. La réponse inclut deux résultats et un jeton qui peut être utilisé lors de la prochaine demande. Lorsqu'une réponse ultérieure n'inclut pas de jeton, tous les services ont été répertoriés.

```
aws apprunner list-services \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "MaxResults": 2
}
```

Sortie :

```
{
    "NextToken":
 "eyJDdXN0b21lckFjY291bnRJZCI6IjI3MDIwNTQwMjg0NSIsIlNlcnZpY2VTdGF0dXNDb2RlIjoiUFJPVklTU9OU
    "ServiceSummaryList": [
        {
            "CreatedAt": "2020-11-20T19:05:25Z",
            "UpdatedAt": "2020-11-23T12:41:37Z",
            "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
            "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
            "ServiceName": "python-app",
            "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
            "Status": "RUNNING"
        },
        {
            "CreatedAt": "2020-11-06T23:15:30Z",
            "UpdatedAt": "2020-11-23T13:21:22Z",
            "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/ab8f94cfe29a460fb8760afd2ee87555",
            "ServiceId": "ab8f94cfe29a460fb8760afd2ee87555",
            "ServiceName": "golang-container-app",
            "ServiceUrl": "e2m8rrrx33.us-east-1.awsapprunner.com",
            "Status": "RUNNING"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListServicesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags associés à un service App Runner

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées à un service App Runner.

```
aws apprunner list-tags-for-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Retail"
        },
        {
            "Key": "CustomerId",
            "Value": "56439872357912"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`pause-service`

L'exemple de code suivant montre comment utiliser`pause-service`.

AWS CLI

Pour suspendre un service

L'`pause-service`exemple suivant met en pause un service App Runner.

```
aws apprunner pause-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-23T12:41:37Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
```

```
                    "StartCommand": "python server.py"
                },
                "ConfigurationSource": "Api"
            },
            "RepositoryUrl": "https://github.com/my-account/python-hello",
            "SourceCodeVersion": {
                "Type": "BRANCH",
                "Value": "main"
            }
        }
    },
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous PauseServiceà la section Référence des AWS CLI commandes.

`resume-service`

L'exemple de code suivant montre comment utiliser`resume-service`.

AWS CLI

Pour reprendre un service

L'`resume-service`exemple suivant reprend un service App Runner.

```
aws apprunner resume-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-23T12:41:37Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
```

```
            "Memory": "3 GB"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ResumeServiceà la section Référence des AWS CLI commandes.

`start-deployment`

L'exemple de code suivant montre comment utiliser`start-deployment`.

AWS CLI

Pour lancer un déploiement manuel

L'`start-deployment`exemple suivant effectue un déploiement manuel vers un service App Runner.

```
aws apprunner start-deployment \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Sortie :

```
{
    "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"
}
```

- Pour plus de détails sur l'API, reportez-vous StartDeploymentà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des tags à un service App Runner

L'`tag-resource`exemple suivant ajoute deux balises à un service App Runner.

```
aws apprunner tag-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "Tags": [
        {
            "Key": "Department",
            "Value": "Retail"
        },
        {
            "Key": "CustomerId",
            "Value": "56439872357912"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des tags d'un service App Runner

L'`untag-resource`exemple suivant supprime deux balises d'un service App Runner.

```
aws apprunner untag-resource \
```

```
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "TagKeys": [
        "Department",
        "CustomerId"
    ]
}
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-service`

L'exemple de code suivant montre comment utiliser`update-service`.

AWS CLI

Pour mettre à jour la taille de la mémoire

L'`update-service`exemple suivant met à jour la taille de mémoire des instances (unités de mise à l'échelle) d'un service App Runner à 2 048 MiB.

Lorsque l'appel aboutit, App Runner lance un processus de mise à jour asynchrone. La `Service` structure renvoyée par l'appel reflète la nouvelle valeur de mémoire appliquée par cet appel.

```
aws apprunner update-service \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "InstanceConfiguration": {
```

```
        "Memory": "4 GB"
    }
}
```

Sortie :

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-23T12:41:37Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
```

```
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "4 GB"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateServiceà la section Référence des AWS CLI commandes.

AWS AppConfig exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS AppConfig.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-application`

L'exemple de code suivant montre comment utiliser`create-application`.

AWS CLI

Pour créer une application

L'`create-application`exemple suivant crée une application dans AWS AppConfig.

```
aws appconfig create-application \
```

```
    --name "example-application" \
    --description "An application used for creating an example."
```

Sortie :

```
{
    "Description": "An application used for creating an example.",
    "Id": "339ohji",
    "Name": "example-application"
}
```

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateApplicationà la section Référence des AWS CLI commandes.

`create-configuration-profile`

L'exemple de code suivant montre comment utiliser`create-configuration-profile`.

AWS CLI

Pour créer un profil de configuration

L'`create-configuration-profile`exemple suivant crée un profil de configuration à l'aide d'une configuration stockée dans Parameter Store, une fonctionnalité de Systems Manager.

```
aws appconfig create-configuration-profile \
    --application-id "339ohji" \
    --name "Example-Configuration-Profile" \
    --location-uri "ssm-parameter://Example-Parameter" \
    --retrieval-role-arn "arn:aws:iam::111122223333:role/Example-App-Config-Role"
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "Description": null,
    "Id": "ur8hx2f",
    "LocationUri": "ssm-parameter://Example-Parameter",
    "Name": "Example-Configuration-Profile",
```

```
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role",
    "Type": null,
    "Validators": null
}
```

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateConfigurationProfileà la section Référence des AWS CLI commandes.

`create-environment`

L'exemple de code suivant montre comment utiliser`create-environment`.

AWS CLI

Pour créer un environnement

L'`create-environment`exemple suivant crée un AWS AppConfig environnement nommé Example-Environment à l'aide de l'application que vous avez créée à l'aide de create-application.

```
aws appconfig create-environment \
    --application-id "339ohji" \
    --name "Example-Environment"
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "Description": null,
    "Id": "54j1r29",
    "Monitors": null,
    "Name": "Example-Environment",
    "State": "ReadyForDeployment"
}
```

Pour plus d'informations, reportez-vous à l'étape 2 : Création d'un environnement dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateEnvironmentà la section Référence des AWS CLI commandes.

`create-extension-association`

L'exemple de code suivant montre comment utiliser`create-extension-association`.

AWS CLI

Pour créer une association d'extensions

L'`create-extension-association`exemple suivant crée une nouvelle association d'extension dans AWS AppConfig.

```
aws appconfig create-extension-association \
    --region us-west-2 \
    --extension-identifier S3-backup-extension \
    --resource-identifier "arn:aws:appconfig:us-west-2:123456789012:application/
Finance" \
    --parameters S3bucket=FinanceConfigurationBackup
```

Sortie :

```
{
  "Id": "a1b2c3d4",
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
  "Parameters": {
    "S3bucket": "FinanceConfigurationBackup"
  },
  "ExtensionVersionNumber": 1
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateExtensionAssociationà la section Référence des AWS CLI commandes.

`create-extension`

L'exemple de code suivant montre comment utiliser`create-extension`.

AWS CLI

Pour créer une extension

L'`create-extension`exemple suivant crée une nouvelle extension dans AWS AppConfig.

```
aws appconfig create-extension \
    --region us-west-2 \
    --name S3-backup-extension \
    --
actions PRE_CREATE_HOSTED_CONFIGURATION_VERSION=[{Name=S3backup,Uri=arn:aws:lambda:us-
west-2:123456789012:function:s3backupfunction,RoleArn=arn:aws:iam::123456789012:role/
appconfigextensionrole}] \
    --parameters S3bucket={Required=true}
```

Sortie :

```
{
    "Id": "1A2B3C4D",
    "Name": "S3-backup-extension",
    "VersionNumber": 1,
    "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
                "Name": "S3backup",
                "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:s3backupfunction",
                "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
            }
        ]
    },
    "Parameters": {
        "S3bucket": {
            "Required": true
        }
    }
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateExtensionà la section Référence des AWS CLI commandes.

`create-hosted-configuration-version`

L'exemple de code suivant montre comment utiliser`create-hosted-configuration-version`.

AWS CLI

Pour créer une version de configuration hébergée

L'`create-hosted-configuration-version`exemple suivant crée une nouvelle configuration dans le magasin de configuration AWS AppConfig hébergé. Le contenu de la configuration doit d'abord être converti en base64.

```
aws appconfig create-hosted-configuration-version \
    --application-id "339ohji" \
    --configuration-profile-id "ur8hx2f" \
    --
content eyAiTmFtZSI6ICJFeGFtcGxlQXBwbGljYXRpb24iLCAiSWQiOiBFeGFtcGxlSUQsICJSYW5rIjogNyB9
 \
    --content-type "application/json" \
    configuration_version_output_file
```

Contenu de `configuration_version_output_file` :

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": "1",
    "ContentType": "application/json"
}
```

Pour plus d'informations, consultez la section À propos du magasin de configuration AWS AppConfig hébergé dans le guide de l'utilisateur d'AWS Appconfig.

- Pour plus de détails sur l'API, reportez-vous CreateHostedConfigurationVersionà la section Référence des AWS CLI commandes.

`delete-application`

L'exemple de code suivant montre comment utiliser`delete-application`.

AWS CLI

Pour supprimer une application

L'`delete-application`exemple suivant supprime l'application spécifiée.

```
aws appconfig delete-application \
--application-id 339ohji
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteApplicationà la section Référence des AWS CLI commandes.

`delete-configuration-profile`

L'exemple de code suivant montre comment utiliser`delete-configuration-profile`.

AWS CLI

Pour supprimer un profil de configuration

L'`delete-configuration-profile`exemple suivant supprime le profil de configuration spécifié.

```
aws appconfig delete-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteConfigurationProfileà la section Référence des AWS CLI commandes.

`delete-deployment-strategy`

L'exemple de code suivant montre comment utiliser`delete-deployment-strategy`.

AWS CLI

Pour supprimer une stratégie de déploiement

L'`delete-deployment-strategy`exemple suivant supprime la stratégie de déploiement spécifiée.

```
aws appconfig delete-deployment-strategy \
    --deployment-strategy-id 1225qzk
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Étape 4 : Création d'une stratégie de déploiement dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteDeploymentStrategyà la section Référence des AWS CLI commandes.

`delete-environment`

L'exemple de code suivant montre comment utiliser`delete-environment`.

AWS CLI

Pour supprimer un environnement

L'`delete-environment`exemple suivant supprime l'environnement d'application spécifié.

```
aws appconfig delete-environment \
    --application-id 339ohji \
    --environment-id 54j1r29
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, reportez-vous à l'étape 2 : Création d'un environnement dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteEnvironmentà la section Référence des AWS CLI commandes.

`delete-extension-association`

L'exemple de code suivant montre comment utiliser`delete-extension-association`.

AWS CLI

Pour supprimer une association d'extensions

L'`delete-extension-association`exemple suivant supprime une association d'extension de AWS AppConfig.

```
aws appconfig delete-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteExtensionAssociationà la section Référence des AWS CLI commandes.

`delete-extension`

L'exemple de code suivant montre comment utiliser`delete-extension`.

AWS CLI

Pour supprimer une extension

L'`delete-extension`exemple suivant supprime une extension de AWS AppConfig.

```
aws appconfig delete-extension \
```

```
    --region us-west-2 \
    --extension-identifier S3-backup-extension
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteExtensionà la section Référence des AWS CLI commandes.

`delete-hosted-configuration-version`

L'exemple de code suivant montre comment utiliser`delete-hosted-configuration-version`.

AWS CLI

Pour supprimer une version de configuration hébergée

L'`delete-hosted-configuration-version`exemple suivant supprime une version de configuration hébergée dans le magasin de configuration AWS AppConfig hébergé.

```
aws appconfig delete-hosted-configuration-version \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --version-number 1
```

Sortie : Cette commande ne produit aucune sortie.

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteHostedConfigurationVersionà la section Référence des AWS CLI commandes.

`get-application`

L'exemple de code suivant montre comment utiliser`get-application`.

AWS CLI

Pour répertorier les détails d'une application

L'`get-application`exemple suivant répertorie les détails de l'application spécifiée.

```
aws appconfig get-application \
    --application-id 339ohji
```

Sortie :

```
{
    "Description": "An application used for creating an example.",
    "Id": "339ohji",
    "Name": "example-application"
}
```

Pour plus d'informations, consultez la section AWS AppConfig Fonctionnement du guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetApplicationà la section Référence des AWS CLI commandes.

`get-configuration-profile`

L'exemple de code suivant montre comment utiliser`get-configuration-profile`.

AWS CLI

Pour récupérer les détails du profil de configuration

L'`get-configuration-profile`exemple suivant renvoie les détails du profil de configuration spécifié.

```
aws appconfig get-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "Id": "ur8hx2f",
    "Name": "Example-Configuration-Profile",
    "LocationUri": "ssm-parameter://Example-Parameter",
```

```
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetConfigurationProfileà la section Référence des AWS CLI commandes.

`get-configuration`

L'exemple de code suivant montre comment utiliser`get-configuration`.

AWS CLI

Pour récupérer les détails de configuration

L'`get-configuration`exemple suivant renvoie les détails de configuration de l'exemple d'application. Lors des appels suivants à get-configuration, utilisez le `client-configuration-version` paramètre pour ne mettre à jour la configuration de votre application que si la version a changé. La mise à jour de la configuration uniquement lorsque la version a changé permet d'éviter les frais supplémentaires liés à l'appel à get-configuration.

```
aws appconfig get-configuration \
    --application "example-application" \
    --environment "Example-Environment" \
    --configuration "Example-Configuration-Profile" \
    --client-id "test-id" \
    configuration-output-file
```

Contenu de `configuration-output-file` :

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Sortie :

```
{
    "ConfigurationVersion": "1",
    "ContentType": "application/json"
}
```

Pour plus d'informations, reportez-vous à l'étape 6 : réception de la configuration dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetConfigurationà la section Référence des AWS CLI commandes.

`get-deployment-strategy`

L'exemple de code suivant montre comment utiliser`get-deployment-strategy`.

AWS CLI

Pour récupérer les détails d'une stratégie de déploiement

L'`get-deployment-strategy`exemple suivant répertorie les détails de la stratégie de déploiement spécifiée.

```
aws appconfig get-deployment-strategy \
    --deployment-strategy-id 1225qzk
```

Sortie :

```
{
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "ReplicateTo": "SSM_DOCUMENT"
}
```

Pour plus d'informations, voir Étape 4 : Création d'une stratégie de déploiement dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetDeploymentStrategyà la section Référence des AWS CLI commandes.

`get-deployment`

L'exemple de code suivant montre comment utiliser`get-deployment`.

AWS CLI

Pour récupérer les détails du déploiement

L'`get-deployment`exemple suivant répertorie les détails du déploiement de l'application dans l'environnement et le déploiement spécifiés.

```
aws appconfig get-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-number 1
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "EnvironmentId": "54j1r29",
    "DeploymentStrategyId": "1225qzk",
    "ConfigurationProfileId": "ur8hx2f",
    "DeploymentNumber": 1,
    "ConfigurationName": "Example-Configuration-Profile",
    "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",
    "ConfigurationVersion": "1",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "State": "COMPLETE",
    "EventLog": [
        {
            "EventType": "DEPLOYMENT_COMPLETED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Deployment completed",
            "OccurredAt": "2021-09-17T21:59:03.888000+00:00"
        },
        {
            "EventType": "BAKE_TIME_STARTED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Deployment bake time started",
            "OccurredAt": "2021-09-17T21:58:57.722000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
```

```
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 100.00% of clients",
            "OccurredAt": "2021-09-17T21:55:56.816000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 75.00% of clients",
            "OccurredAt": "2021-09-17T21:52:56.567000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 50.00% of clients",
            "OccurredAt": "2021-09-17T21:49:55.737000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 25.00% of clients",
            "OccurredAt": "2021-09-17T21:46:55.187000+00:00"
        },
        {
            "EventType": "DEPLOYMENT_STARTED",
            "TriggeredBy": "USER",
            "Description": "Deployment started",
            "OccurredAt": "2021-09-17T21:43:54.205000+00:00"
        }
    ],
    "PercentageComplete": 100.0,
    "StartedAt": "2021-09-17T21:43:54.205000+00:00",
    "CompletedAt": "2021-09-17T21:59:03.888000+00:00"
}
```

Pour plus d'informations, voir Étape 5 : Déploiement d'une configuration dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetDeploymentà la section Référence des AWS CLI commandes.

`get-environment`

L'exemple de code suivant montre comment utiliser`get-environment`.

AWS CLI

Pour récupérer les détails de l'environnement

L'`get-environment`exemple suivant renvoie les détails et l'état de l'environnement spécifié.

```
aws appconfig get-environment \
    --application-id 339ohji \
    --environment-id 54j1r29
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "Id": "54j1r29",
    "Name": "Example-Environment",
    "State": "ReadyForDeployment"
}
```

Pour plus d'informations, reportez-vous à l'étape 2 : Création d'un environnement dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetEnvironmentà la section Référence des AWS CLI commandes.

`get-extension-association`

L'exemple de code suivant montre comment utiliser`get-extension-association`.

AWS CLI

Pour obtenir les détails de l'association d'extensions

L'`get-extension-association`exemple suivant affiche des informations sur une association d'extensions.

```
aws appconfig get-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4
```

Sortie :

```
{
    "Id": "a1b2c3d4",
    "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
    "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
    "Parameters": {
        "S3bucket": "FinanceConfigurationBackup"
    },
    "ExtensionVersionNumber": 1
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetExtensionAssociationà la section Référence des AWS CLI commandes.

`get-extension`

L'exemple de code suivant montre comment utiliser`get-extension`.

AWS CLI

Pour obtenir les détails de l'extension

L'`get-extension`exemple suivant affiche des informations sur une extension.

```
aws appconfig get-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension
```

Sortie :

```
{
    "Id": "1A2B3C4D",
    "Name": "S3-backup-extension",
    "VersionNumber": 1,
    "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
```

```
                "Name": "S3backup",
                "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:S3backupfunction",
                "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
            }
        ]
    },
    "Parameters": {
        "S3bucket": {
            "Required": true
        }
    }
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetExtensionà la section Référence des AWS CLI commandes.

`get-hosted-configuration-version`

L'exemple de code suivant montre comment utiliser`get-hosted-configuration-version`.

AWS CLI

Pour récupérer les détails de la configuration hébergée

L'`get-hosted-configuration-version`exemple suivant récupère les détails de configuration de la configuration AWS AppConfig hébergée.

```
aws appconfig get-hosted-configuration-version \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --version-number 1 \
    hosted-configuration-version-output
```

Contenu de `hosted-configuration-version-output` :

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": "1",
    "ContentType": "application/json"
}
```

Pour plus d'informations, consultez la section À propos du magasin de configuration AWS AppConfig hébergé dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetHostedConfigurationVersionà la section Référence des AWS CLI commandes.

`list-applications`

L'exemple de code suivant montre comment utiliser`list-applications`.

AWS CLI

Pour répertorier les applications disponibles

L'`list-applications`exemple suivant répertorie les applications disponibles dans votre AWS compte.

```
aws appconfig list-applications
```

Sortie :

```
{
    "Items": [
        {
            "Id": "339ohji",
            "Name": "test-application",
            "Description": "An application used for creating an example."
        },
        {
            "Id": "rwalwu7",
            "Name": "Test-Application"
        }
    ]
}
```

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListApplicationsà la section Référence des AWS CLI commandes.

`list-configuration-profiles`

L'exemple de code suivant montre comment utiliser`list-configuration-profiles`.

AWS CLI

Pour répertorier les profils de configuration disponibles

L'`list-configuration-profiles`exemple suivant répertorie les profils de configuration disponibles pour l'application spécifiée.

```
aws appconfig list-configuration-profiles \
    --application-id 339ohji
```

Sortie :

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "Id": "ur8hx2f",
            "Name": "Example-Configuration-Profile",
            "LocationUri": "ssm-parameter://Example-Parameter"
        }
    ]
}
```

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListConfigurationProfilesà la section Référence des AWS CLI commandes.

`list-deployment-strategies`

L'exemple de code suivant montre comment utiliser`list-deployment-strategies`.

AWS CLI

Pour répertorier les stratégies de déploiement disponibles

L'`list-deployment-strategies`exemple suivant répertorie les stratégies de déploiement disponibles dans votre AWS compte.

```
aws appconfig list-deployment-strategies
```

Sortie :

```
{
    "Items": [
        {
            "Id": "1225qzk",
            "Name": "Example-Deployment",
            "DeploymentDurationInMinutes": 15,
            "GrowthType": "LINEAR",
            "GrowthFactor": 25.0,
            "FinalBakeTimeInMinutes": 0,
            "ReplicateTo": "SSM_DOCUMENT"
        },
        {
            "Id": "AppConfig.AllAtOnce",
            "Name": "AppConfig.AllAtOnce",
            "Description": "Quick",
            "DeploymentDurationInMinutes": 0,
            "GrowthType": "LINEAR",
            "GrowthFactor": 100.0,
            "FinalBakeTimeInMinutes": 10,
            "ReplicateTo": "NONE"
        },
        {
            "Id": "AppConfig.Linear50PercentEvery30Seconds",
            "Name": "AppConfig.Linear50PercentEvery30Seconds",
            "Description": "Test/Demo",
            "DeploymentDurationInMinutes": 1,
            "GrowthType": "LINEAR",
            "GrowthFactor": 50.0,
            "FinalBakeTimeInMinutes": 1,
            "ReplicateTo": "NONE"
        },
        {
```

```
            "Id": "AppConfig.Canary10Percent20Minutes",
            "Name": "AppConfig.Canary10Percent20Minutes",
            "Description": "AWS Recommended",
            "DeploymentDurationInMinutes": 20,
            "GrowthType": "EXPONENTIAL",
            "GrowthFactor": 10.0,
            "FinalBakeTimeInMinutes": 10,
            "ReplicateTo": "NONE"
        }
    ]
}
```

Pour plus d'informations, voir Étape 4 : Création d'une stratégie de déploiement dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListDeploymentStrategiesà la section Référence des AWS CLI commandes.

`list-deployments`

L'exemple de code suivant montre comment utiliser`list-deployments`.

AWS CLI

Pour répertorier les déploiements disponibles

L'`list-deployments`exemple suivant répertorie les déploiements disponibles dans votre AWS compte pour l'application et l'environnement spécifiés.

```
aws appconfig list-deployments \
    --application-id 339ohji \
    --environment-id 54j1r29
```

Sortie :

```
{
    "Items": [
        {
            "DeploymentNumber": 1,
            "ConfigurationName": "Example-Configuration-Profile",
            "ConfigurationVersion": "1",
            "DeploymentDurationInMinutes": 15,
```

```
            "GrowthType": "LINEAR",
            "GrowthFactor": 25.0,
            "FinalBakeTimeInMinutes": 0,
            "State": "COMPLETE",
            "PercentageComplete": 100.0,
            "StartedAt": "2021-09-17T21:43:54.205000+00:00",
            "CompletedAt": "2021-09-17T21:59:03.888000+00:00"
        }
    ]
}
```

Pour plus d'informations, voir Étape 5 : Déploiement d'une configuration dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListDeploymentsà la section Référence des AWS CLI commandes.

`list-environments`

L'exemple de code suivant montre comment utiliser`list-environments`.

AWS CLI

Pour répertorier les environnements disponibles

L'`list-environments`exemple suivant répertorie les environnements disponibles dans votre AWS compte pour l'application spécifiée.

```
aws appconfig list-environments \
    --application-id 339ohji
```

Sortie :

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "Id": "54j1r29",
            "Name": "Example-Environment",
            "State": "ReadyForDeployment"
        }
    ]
```

```
}
```

Pour plus d'informations, reportez-vous à l'étape 2 : Création d'un environnement dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListEnvironmentsà la section Référence des AWS CLI commandes.

`list-extension-associations`

L'exemple de code suivant montre comment utiliser`list-extension-associations`.

AWS CLI

Pour répertorier toutes les associations d' AWS AppConfig extensions de votre AWS compte pour une AWS région

L'`list-extension-associations`exemple suivant répertorie toutes les associations d' AWS AppConfig extensions pour le AWS compte courant dans une AWS région spécifique.

```
aws appconfig list-extension-associations \
    --region us-west-2
```

Sortie :

```
{
    "Items": [
        {
            "Id": "a1b2c3d4",
            "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-
backup-extension/1",
            "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/
Finance"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListExtensionAssociationsà la section Référence des AWS CLI commandes.

`list-extensions`

L'exemple de code suivant montre comment utiliser`list-extensions`.

AWS CLI

Pour répertorier toutes les AWS AppConfig extensions de votre AWS compte pour une AWS région

L'`list-extensions`exemple suivant répertorie toutes les AWS AppConfig extensions du AWS compte courant dans une AWS région spécifique. La commande renvoie les extensions personnalisées AWS et créées.

```
aws appconfig list-extensions \
    --region us-west-2
```

Sortie :

```
{
    "Items": [
        {
            "Id": "1A2B3C4D",
            "Name": "S3-backup-extension",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1"
        },
        {
            "Id": "AWS.AppConfig.FeatureFlags",
            "Name": "AppConfig Feature Flags Helper",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.FeatureFlags/1",
            "Description": "Validates AppConfig feature flag data automatically
 against a JSON schema that includes structure and constraints. Also transforms
 feature flag data prior to sending to the client. This extension is automatically
 associated to configuration profiles with type \"AWS.AppConfig.FeatureFlags\"."
        },
        {
            "Id": "AWS.AppConfig.JiraIntegration",
            "Name": "AppConfig integration with Atlassian Jira",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.JiraIntegration/1",
```

```
            "Description": "Exports feature flag data from AWS AppConfig into
 Jira. The lifecycle of each feature flag in AppConfig is tracked in Jira as an
 individual issue. Customers can see in Jira when flags are updated, turned on or
 off. Works in conjunction with the AppConfig app in the Atlassian Marketplace and
 is automatically associated to configuration profiles configured within that app."
        },
        {
            "Id": "AWS.AppConfig.DeploymentNotificationsToEventBridge",
            "Name": "AppConfig deployment events to Amazon EventBridge",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToEventBridge/1",
            "Description": "Sends events to Amazon EventBridge when a deployment
 of configuration data in AppConfig is started, completed, or rolled back. Can
 be associated to the following resources in AppConfig: Application, Environment,
 Configuration Profile."
        },
        {
            "Id": "AWS.AppConfig.DeploymentNotificationsToSqs",
            "Name": "AppConfig deployment events to Amazon SQS",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToSqs/1",
            "Description": "Sends messages to the configured Amazon SQS queue when
 a deployment of configuration data in AppConfig is started, completed, or rolled
 back. Can be associated to the following resources in AppConfig: Application,
 Environment, Configuration Profile."
        },
        {
            "Id": "AWS.AppConfig.DeploymentNotificationsToSns",
            "Name": "AppConfig deployment events to Amazon SNS",
            "VersionNumber": 1,
            "Description": "Sends events to the configured Amazon SNS topic when
 a deployment of configuration data in AppConfig is started, completed, or rolled
 back. Can be associated to the following resources in AppConfig: Application,
 Environment, Configuration Profile."
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListExtensionsà la section Référence des AWS CLI commandes.

`list-hosted-configuration-versions`

L'exemple de code suivant montre comment utiliser`list-hosted-configuration-versions`.

AWS CLI

Pour répertorier les versions de configuration hébergées disponibles

L'`list-hosted-configuration-versions`exemple suivant répertorie les versions de configurations hébergées dans le magasin de configuration AWS AppConfig hébergé pour l'application et le profil de configuration spécifiés.

```
aws appconfig list-hosted-configuration-versions \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

Sortie :

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "ConfigurationProfileId": "ur8hx2f",
            "VersionNumber": 1,
            "ContentType": "application/json"
        }
    ]
}
```

Pour plus d'informations, consultez la section À propos du magasin de configuration AWS AppConfig hébergé dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListHostedConfigurationVersionsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une application

L'`list-tags-for-resource`exemple suivant répertorie les balises d'une application spécifiée.

```
aws appconfig list-tags-for-resource \
    --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

Sortie :

```
{
    "Tags": {
        "group1": "1"
    }
}
```

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-deployment`

L'exemple de code suivant montre comment utiliser`start-deployment`.

AWS CLI

Pour démarrer un déploiement de configuration

L'`start-deployment`exemple suivant lance un déploiement vers l'application en utilisant l'environnement, la stratégie de déploiement et le profil de configuration spécifiés.

```
aws appconfig start-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-strategy-id 1225qzk \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "EnvironmentId": "54j1r29",
    "DeploymentStrategyId": "1225qzk",
    "ConfigurationProfileId": "ur8hx2f",
    "DeploymentNumber": 1,
    "ConfigurationName": "Example-Configuration-Profile",
    "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",
    "ConfigurationVersion": "1",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "State": "DEPLOYING",
    "EventLog": [
        {
            "EventType": "DEPLOYMENT_STARTED",
            "TriggeredBy": "USER",
            "Description": "Deployment started",
            "OccurredAt": "2021-09-17T21:43:54.205000+00:00"
        }
    ],
    "PercentageComplete": 0.0,
    "StartedAt": "2021-09-17T21:43:54.205000+00:00"
}
```

Pour plus d'informations, voir Étape 5 : Déploiement d'une configuration dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StartDeploymentà la section Référence des AWS CLI commandes.

`stop-deployment`

L'exemple de code suivant montre comment utiliser`stop-deployment`.

AWS CLI

Pour arrêter le déploiement de la configuration

L'`stop-deployment`exemple suivant arrête le déploiement d'une configuration d'application dans l'environnement spécifié.

```
aws appconfig stop-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-number 2
```

Sortie :

```
{
    "DeploymentNumber": 0,
    "DeploymentDurationInMinutes": 0,
    "GrowthFactor": 0.0,
    "FinalBakeTimeInMinutes": 0,
    "PercentageComplete": 0.0
}
```

Pour plus d'informations, voir Étape 5 : Déploiement d'une configuration dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StopDeploymentà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une application

L'`tag-resource`exemple suivant balise une ressource d'application.

```
aws appconfig tag-resource \
    --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \
    --tags '{"group1" : "1"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une application

L'`untag-resource`exemple suivant supprime la balise group1 de l'application spécifiée.

```
aws appconfig untag-resource \
    --resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \
    --tag-keys '["group1"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-application`

L'exemple de code suivant montre comment utiliser`update-application`.

AWS CLI

Pour mettre à jour une application

L'`update-application`exemple suivant met à jour le nom de l'application spécifiée.

```
aws appconfig update-application \
    --application-id 339ohji \
    --name "Example-Application"
```

Sortie :

```
{
    "Id": "339ohji",
    "Name": "Example-Application",
    "Description": "An application used for creating an example."
}
```

Pour plus d'informations, voir Étape 1 : Création d'une AWS AppConfig application dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateApplicationà la section Référence des AWS CLI commandes.

`update-configuration-profile`

L'exemple de code suivant montre comment utiliser`update-configuration-profile`.

AWS CLI

Pour mettre à jour un profil de configuration

L'`update-configuration-profile`exemple suivant met à jour la description du profil de configuration spécifié.

```
aws appconfig update-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --description "Configuration profile used for examples."
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "Id": "ur8hx2f",
    "Name": "Example-Configuration-Profile",
    "Description": "Configuration profile used for examples.",
    "LocationUri": "ssm-parameter://Example-Parameter",
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateConfigurationProfileà la section Référence des AWS CLI commandes.

`update-deployment-strategy`

L'exemple de code suivant montre comment utiliser`update-deployment-strategy`.

AWS CLI

Pour mettre à jour une stratégie de déploiement

L'`update-deployment-strategy`exemple suivant met à jour le temps de cuisson final à 20 minutes dans la stratégie de déploiement spécifiée.

```
aws appconfig update-deployment-strategy \
    --deployment-strategy-id 1225qzk \
    --final-bake-time-in-minutes 20
```

Sortie :

```
{
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 20,
    "ReplicateTo": "SSM_DOCUMENT"
}
```

Pour plus d'informations, voir Étape 4 : Création d'une stratégie de déploiement dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateDeploymentStrategyà la section Référence des AWS CLI commandes.

`update-environment`

L'exemple de code suivant montre comment utiliser`update-environment`.

AWS CLI

Pour mettre à jour un environnement

L'`update-environment`exemple suivant met à jour la description d'un environnement.

```
aws appconfig update-environment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
```

```
    --description "An environment for examples."
```

Sortie :

```
{
    "ApplicationId": "339ohji",
    "Id": "54j1r29",
    "Name": "Example-Environment",
    "Description": "An environment for examples.",
    "State": "RolledBack"
}
```

Pour plus d'informations, reportez-vous à l'étape 2 : Création d'un environnement dans le guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateEnvironmentà la section Référence des AWS CLI commandes.

`update-extension-association`

L'exemple de code suivant montre comment utiliser`update-extension-association`.

AWS CLI

Pour mettre à jour une association d' AWS AppConfig extensions

L'`update-extension-association`exemple suivant ajoute une nouvelle valeur de paramètre à une association d'extension dans AWS AppConfig.

```
aws appconfig update-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4 \
    --parameters S3bucket=FinanceMobileApp
```

Sortie :

```
{
    "Id": "a1b2c3d4",
    "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
    "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
```

```
    "Parameters": {
        "S3bucket": "FinanceMobileApp"
    },
    "ExtensionVersionNumber": 1
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateExtensionAssociationà la section Référence des AWS CLI commandes.

`update-extension`

L'exemple de code suivant montre comment utiliser`update-extension`.

AWS CLI

Pour mettre à jour une AWS AppConfig extension

L'`update-extension`exemple suivant ajoute un paramètre supplémentaire Key à une extension dans AWS AppConfig.

```
aws appconfig update-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension \
    --parameters S3bucket={Required=true},CampaignID={Required=false}
```

Sortie :

```
{
    "Id": "1A2B3C4D",
    "Name": "S3-backup-extension",
    "VersionNumber": 1,
    "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
                "Name": "S3backup",
                "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:S3backupfunction",
                "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
```

```
            }
        ]
    },
    "Parameters": {
        "CampaignID": {
            "Required": false
        },
        "S3bucket": {
            "Required": true
        }
    }
}
```

Pour plus d'informations, consultez la section Utilisation des AWS AppConfig extensions dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateExtensionà la section Référence des AWS CLI commandes.

`validate-configuration`

L'exemple de code suivant montre comment utiliser`validate-configuration`.

AWS CLI

Pour valider une configuration

L'`validate-configuration`exemple suivant utilise les validateurs d'un profil de configuration pour valider une configuration.

```
aws appconfig validate-configuration \
    --application-id abc1234 \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

La commande ne produit aucune sortie.

Pour plus d'informations, voir Étape 3 : Création d'une configuration et d'un profil de configuration dans le Guide de AWS AppConfig l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ValidateConfigurationà la section Référence des AWS CLI commandes.

Exemples d'Application Auto Scaling utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l' AWS Command Line Interface application Application Auto Scaling.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-scaling-policy`

L'exemple de code suivant montre comment utiliser`delete-scaling-policy`.

AWS CLI

Pour supprimer une politique de dimensionnement

Cet exemple supprime une politique de dimensionnement pour l'application Web du service Amazon ECS exécutée dans le cluster par défaut.

Commande :

```
aws application-autoscaling delete-scaling-policy --policy-name web-app-cpu-lt-25 --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app --
service-namespace ecs
```

- Pour plus de détails sur l'API, reportez-vous DeleteScalingPolicyà la section Référence des AWS CLI commandes.

`delete-scheduled-action`

L'exemple de code suivant montre comment utiliser`delete-scheduled-action`.

AWS CLI

Pour supprimer une action planifiée

L'`delete-scheduled-action`exemple suivant supprime l'action planifiée spécifiée de la flotte Amazon AppStream 2.0 spécifiée :

```
aws application-autoscaling delete-scheduled-action \
    --service-namespace appstream \
    --scalable-dimension appstream:fleet:DesiredCapacity \
    --resource-id fleet/sample-fleet \
    --scheduled-action-name my-recurring-action
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Mise à l'échelle planifiée dans le Guide de l'utilisateur Application Auto Scaling..

- Pour plus de détails sur l'API, reportez-vous DeleteScheduledActionà la section Référence des AWS CLI commandes.

`deregister-scalable-target`

L'exemple de code suivant montre comment utiliser`deregister-scalable-target`.

AWS CLI

Pour annuler l'enregistrement d'une cible évolutive

Cet exemple annule l'enregistrement d'une cible évolutive pour un service Amazon ECS appelé application Web qui s'exécute dans le cluster par défaut.

Commande :

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

Cet exemple désenregistre une cible évolutive pour une ressource personnalisée. Le fichier custom-resource-id .txt contient une chaîne qui identifie l'ID de ressource, qui, pour une ressource personnalisée, est le chemin d'accès à la ressource personnalisée via votre point de terminaison Amazon API Gateway.

Commande :

```
aws application-autoscaling deregister-scalable-target --service-namespace custom-resource --scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

Contenu du fichier custom-resource-id .txt :

```
https://example.execute-api.us-west-2.amazonaws.com/prod/scalableTargetDimensions/1-23456789
```

- Pour plus de détails sur l'API, reportez-vous DeregisterScalableTargetà la section Référence des AWS CLI commandes.

`describe-scalable-targets`

L'exemple de code suivant montre comment utiliser`describe-scalable-targets`.

AWS CLI

Pour décrire les cibles évolutives

L'`describe-scalable-targets`exemple suivant décrit les cibles évolutives pour l'espace de noms `ecs` de service.

```
aws application-autoscaling describe-scalable-targets \
    --service-namespace ecs
```

Sortie :

```
{
    "ScalableTargets": [
        {
            "ServiceNamespace": "ecs",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "MinCapacity": 1,
            "MaxCapacity": 10,
            "RoleARN": "arn:aws:iam::123456789012:role/aws-service-role/ecs.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService",
            "CreationTime": 1462558906.199,
```

```
            "SuspendedState": {
                "DynamicScalingOutSuspended": false,
                "ScheduledScalingSuspended": false,
                "DynamicScalingInSuspended": false
            },
            "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
        }
    ]
}
```

Pour plus d'informations, consultez les AWS services que vous pouvez utiliser avec Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalableTargetsà la section Référence des AWS CLI commandes.

`describe-scaling-activities`

L'exemple de code suivant montre comment utiliser`describe-scaling-activities`.

AWS CLI

Exemple 1 : pour décrire les activités de dimensionnement pour le service Amazon ECS spécifié

L'`describe-scaling-activities`exemple suivant décrit les activités de dimensionnement d'un service Amazon ECS appelé `web-app` qui s'exécute dans le `default` cluster. Le résultat montre une activité de dimensionnement initiée par une politique de dimensionnement.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace ecs \
    --resource-id service/default/web-app
```

Sortie :

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Setting desired count to 1.",
            "ResourceId": "service/default/web-app",
            "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
```

```
            "StartTime": 1462575838.171,
            "ServiceNamespace": "ecs",
            "EndTime": 1462575872.111,
            "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered
 policy web-app-cpu-lt-25",
            "StatusMessage": "Successfully set desired count to 1. Change
 successfully fulfilled by ecs.",
            "StatusCode": "Successful"
        }
    ]
}
```

Pour plus d'informations, consultez la section Activités de dimensionnement pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

Exemple 2 : pour décrire les activités de dimensionnement pour la table DynamoDB spécifiée

L'`describe-scaling-activities`exemple suivant décrit les activités de dimensionnement pour une table DynamoDB appelée. `TestTable` Le résultat montre les activités de dimensionnement initiées par deux actions planifiées différentes.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace dynamodb \
    --resource-id table/TestTable
```

Sortie :

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/my-table",
            "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",
            "StartTime": 1561574415.086,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574449.51,
            "Cause": "maximum capacity was set to 10",
            "StatusMessage": "Successfully set write capacity units to 10. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
```

```
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 5 and max capacity to 10",
            "ResourceId": "table/my-table",
            "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",
            "StartTime": 1561574414.644,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-second-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 5 and max capacity to
 10",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/my-table",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574140.255,
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/my-table",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max capacity
 to 20",
            "StatusCode": "Successful"
        }
    ]
}
```


Pour plus d'informations, consultez la section Activités de dimensionnement pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalingActivitiesà la section Référence des AWS CLI commandes.

`describe-scaling-policies`

L'exemple de code suivant montre comment utiliser`describe-scaling-policies`.

AWS CLI

Pour décrire les politiques de dimensionnement

Cet exemple de commande décrit les politiques de dimensionnement pour l'espace de noms du service ecs.

Commande :

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

Sortie :

```
{
    "ScalingPolicies": [
        {
            "PolicyName": "web-app-cpu-gt-75",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462561899.23,
            "StepScalingPolicyConfiguration": {
                "Cooldown": 60,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": 200,
                        "MetricIntervalLowerBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-gt-75",
            "PolicyType": "StepScaling",
```

```
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-gt-75",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-gt-75"
                }
            ],
            "ServiceNamespace": "ecs"
        },
        {
            "PolicyName": "web-app-cpu-lt-25",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462562575.099,
            "StepScalingPolicyConfiguration": {
                "Cooldown": 1,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": -50,
                        "MetricIntervalUpperBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-lt-25",
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-lt-25",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-lt-25"
                }
            ],
            "ServiceNamespace": "ecs"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeScalingPoliciesà la section Référence des AWS CLI commandes.

`describe-scheduled-actions`

L'exemple de code suivant montre comment utiliser`describe-scheduled-actions`.

AWS CLI

Pour décrire les actions planifiées

L'`describe-scheduled-actions`exemple suivant affiche les détails des actions planifiées pour l'espace de noms de service spécifié :

```
aws application-autoscaling describe-scheduled-actions \
    --service-namespace dynamodb
```

Sortie :

```
{
    "ScheduledActions": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:35:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571888.361,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-first-scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 15,
                "MaxCapacity": 20
            },
            "ScheduledActionName": "my-first-scheduled-action",
            "ServiceNamespace": "dynamodb"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:40:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571946.021,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-second-scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 5,
```

```
                "MaxCapacity": 10
            },
            "ScheduledActionName": "my-second-scheduled-action",
            "ServiceNamespace": "dynamodb"
        }
    ]
}
```

Pour plus d'informations, consultez Mise à l'échelle planifiée dans le Guide de l'utilisateur Application Auto Scaling..

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledActionsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une cible évolutive

L'`list-tags-for-resource`exemple suivant répertorie les noms et les valeurs des clés de balise associées à la cible évolutive spécifiée par son ARN.

```
aws application-autoscaling list-tags-for-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123
```

Sortie :

```
{
    "Tags": {
        "environment": "production"
    }
}
```

Pour plus d'informations, consultez la section Support du balisage pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-scaling-policy`

L'exemple de code suivant montre comment utiliser`put-scaling-policy`.

AWS CLI

Exemple 1 : pour appliquer une politique de suivi des objectifs et d'échelonnement avec une spécification de métrique prédéfinie

L'`put-scaling-policy`exemple suivant applique une politique de dimensionnement du suivi des cibles avec une spécification métrique prédéfinie à un service Amazon ECS appelé web-app dans le cluster par défaut. La politique maintient l'utilisation moyenne du processeur du service à 75 %, avec des périodes d'évolutivité et de recharge de 60 secondes. La sortie contient les noms ARNs et les noms des deux CloudWatch alarmes créées en votre nom.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cpu75-target-tracking-scaling-policy --policy-
type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

Cet exemple suppose que le répertoire en cours contient un fichier config.json dont le contenu est le suivant :

```
{
     "TargetValue": 75.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

Sortie :

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/cpu75-
target-tracking-scaling-policy",
    "Alarms": [
```

```
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d",
            "AlarmName": "TargetTracking-service/default/web-app-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d"
        }
    ]
}
```

Exemple 2 : pour appliquer une politique de suivi des objectifs et d'échelonnement avec une spécification de métrique personnalisée

L'`put-scaling-policy`exemple suivant applique une politique de dimensionnement du suivi des cibles avec une spécification métrique personnalisée à un service Amazon ECS appelé web-app dans le cluster par défaut. La politique maintient le taux d'utilisation moyen du service à 75 %, avec des périodes d'évolutivité et de recharge de 60 secondes. La sortie contient les noms ARNs et les noms des deux CloudWatch alarmes créées en votre nom.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cms75-target-tracking-scaling-policy
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

Cet exemple suppose que le répertoire en cours contient un fichier config.json dont le contenu est le suivant :

```
{
    "TargetValue":75.0,
    "CustomizedMetricSpecification":{
        "MetricName":"MyUtilizationMetric",
        "Namespace":"MyNamespace",
        "Dimensions": [
```

```
            {
                "Name":"MyOptionalMetricDimensionName",
                "Value":"MyOptionalMetricDimensionValue"
            }
        ],
        "Statistic":"Average",
        "Unit":"Percent"
    },
    "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

Sortie :

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
 8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/default/web-
app:policyName/cms75-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
        }
    ]
}
```

Exemple 3 : pour appliquer une politique de suivi des objectifs et d'échelonnement uniquement en vue d'une évolutivité horizontale

L'`put-scaling-policy`exemple suivant applique une politique de dimensionnement du suivi des cibles à un service Amazon ECS appelé `web-app` dans le cluster par défaut. La politique est utilisée pour étendre le service ECS lorsque la `RequestCountPerTarget` métrique de

l'Application Load Balancer dépasse le seuil. La sortie contient l'ARN et le nom de l' CloudWatch alarme créée en votre nom.

```
aws application-autoscaling put-scaling-policy \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --policy-name alb-scale-out-target-tracking-scaling-policy \
    --policy-type TargetTrackingScaling \
    --target-tracking-scaling-policy-configuration file://config.json
```

Contenu de `config.json` :

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ALBRequestCountPerTarget",
         "ResourceLabel": "app/EC2Co-EcsEl-1TKLTMITMM0EO/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60,
    "DisableScaleIn": true
}
```

Sortie :

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/alb-
scale-out-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca"
        }
    ]
}
```

Pour plus d'informations, consultez Target Tracking Scaling Policies for Application Auto Scaling dans le guide de l'utilisateur d'AWS Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutScalingPolicyà la section Référence des AWS CLI commandes.

`put-scheduled-action`

L'exemple de code suivant montre comment utiliser`put-scheduled-action`.

AWS CLI

Pour ajouter une action planifiée à une table DynamoDB

Cet exemple ajoute une action planifiée à une table DynamoDB TestTable appelée pour effectuer une mise à l'échelle selon un calendrier récurrent. Selon le calendrier spécifié (tous les jours à 12 h 15 UTC), si la capacité actuelle est inférieure à la valeur spécifiée pour MinCapacity, Application Auto Scaling s'adapte à la valeur spécifiée par MinCapacity.

Commande :

```
aws application-autoscaling put-scheduled-action --service-namespace dynamodb
 --scheduled-action-name my-recurring-action --schedule "cron(15 12 * * ? *)" --
resource-id table/TestTable --scalable-dimension dynamodb:table:WriteCapacityUnits
 --scalable-target-action MinCapacity=6
```

Pour plus d'informations, consultez la section Scheduled Scaling dans le guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutScheduledActionà la section Référence des AWS CLI commandes.

`register-scalable-target`

L'exemple de code suivant montre comment utiliser`register-scalable-target`.

AWS CLI

Exemple 1 : pour enregistrer un service ECS en tant que cible évolutive

L'`register-scalable-target`exemple suivant enregistre un service Amazon ECS auprès d'Application Auto Scaling. Il ajoute également une balise avec le nom de la clé `environment` et la valeur `production` à la cible évolutive.

```
aws application-autoscaling register-scalable-target \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --min-capacity 1 --max-capacity 10 \
    --tags environment=production
```

Sortie :

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Pour des exemples d'autres AWS services et ressources personnalisées, consultez les rubriques relatives AWS aux services que vous pouvez utiliser avec Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

Exemple 2 : pour suspendre les activités de dimensionnement pour une cible évolutive

L'`register-scalable-target`exemple suivant suspend les activités de dimensionnement pour une cible évolutive existante.

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=true,DynamicScalingOutSuspended=true,ScheduledScalingSuspend
```

Sortie :

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Pour plus d'informations, voir Suspendre et reprendre le dimensionnement pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

Exemple 3 : pour reprendre les activités de dimensionnement pour une cible évolutive

L'`register-scalable-target`exemple suivant reprend les activités de dimensionnement pour une cible évolutive existante.

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=false,DynamicScalingOutSuspended=false,ScheduledScalingSuspe
```

Sortie :

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Pour plus d'informations, voir Suspendre et reprendre le dimensionnement pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous RegisterScalableTargetà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une balise à une cible évolutive

L'`tag-resource`exemple suivant ajoute une balise avec le nom de la clé `environment` et la valeur `production` à la cible évolutive spécifiée par son ARN.

```
aws application-autoscaling tag-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123 \
```

```
    --tags environment=production
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Support du balisage pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une balise d'une cible évolutive

L'`untag-resource`exemple suivant supprime la paire de balises portant le nom `environment` de la clé de la cible évolutive spécifiée par son ARN.

```
aws application-autoscaling untag-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123 \
    --tag-keys "environment"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Support du balisage pour Application Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples d'Application Discovery Service utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de l'Application Discovery Service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-agents`

L'exemple de code suivant montre comment utiliser`describe-agents`.

AWS CLI

Décrire les agents avec des états CollectionStatus spécifiés

Cet exemple de commande décrit les agents de collecte dont le statut de collecte est « DÉMARRÉ » ou « ARRÊTÉ ».

Commande :

```
aws discovery describe-agents --filters
 name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
results 3
```

Sortie :

```
{
       "Snapshots": [
      {
            "version": "1.0.40.0",
            "agentType": "EC2",
            "hostName": "ip-172-31-40-234",
            "collectionStatus": "STOPPED",
            "agentNetworkInfoList": [
                {
                    "macAddress": "06:b5:97:14:fc:0d",
                    "ipAddress": "172.31.40.234"
                }
            ],
            "health": "UNKNOWN",
```

```
            "agentId": "i-003305c02a776e883",
            "registeredTime": "2016-12-09T19:05:06Z",
            "lastHealthPingTime": "2016-12-09T19:05:10Z"
        },
        {
            "version": "1.0.40.0",
            "agentType": "EC2",
            "hostName": "ip-172-31-39-64",
            "collectionStatus": "STARTED",
            "agentNetworkInfoList": [
                {
                    "macAddress": "06:a1:0e:c7:b2:73",
                    "ipAddress": "172.31.39.64"
                }
            ],
            "health": "SHUTDOWN",
            "agentId": "i-003a5e5e2b36cf8bd",
            "registeredTime": "2016-11-16T16:36:25Z",
            "lastHealthPingTime": "2016-11-16T16:47:37Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAgentsà la section Référence des AWS CLI commandes.

`describe-configurations`

L'exemple de code suivant montre comment utiliser`describe-configurations`.

AWS CLI

Décrire les configurations d'actifs sélectionnées

Cet exemple de commande décrit les configurations de deux serveurs spécifiés. L'action détecte le type de ressource à partir de l'ID de configuration. Un seul type de ressource est autorisé par commande.

Commande :

```
aws discovery describe-configurations --configuration-ids "d-
server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

Sortie :


```
{
    "configurations": [
        {
                "server.performance.maxCpuUsagePct": "0.0",
                "server.performance.maxDiskReadIOPS": "0.0",
                "server.performance.avgCpuUsagePct": "0.0",
                "server.type": "EC2",
                "server.performance.maxNetworkReadsPerSecondInKB": "0.19140625",
                "server.hostName": "ip-172-31-35-152",
                "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
                "server.tags.hasMoreValues": "false",
                "server.performance.minFreeRAMInKB": "1543496.0",
                "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
                "server.performance.maxDiskReadsPerSecondInKB": "0.0",
                "server.applications": "[]",
                "server.performance.numDisks": "1",
                "server.performance.numCpus": "1",
                "server.performance.numCores": "1",
                "server.performance.maxDiskWriteIOPS": "0.0",
                "server.performance.maxNetworkWritesPerSecondInKB": "0.82421875",
                "server.performance.avgDiskWritesPerSecondInKB": "0.0",
                "server.networkInterfaceInfo": "[{\"name\":\"eth0\",
\"macAddress\":\"06:A7:7D:3F:54:57\",\"ipAddress\":\"172.31.35.152\",\"netMask\":
\"255.255.240.0\"},{\"name\":\"lo\",\"macAddress\":\"00:00:00:00:00:00\",\"ipAddress
\":\"127.0.0.1\",\"netMask\":\"255.0.0.0\"},{\"name\":\"eth0\",\"macAddress\":
\"06:A7:7D:3F:54:57\",\"ipAddress\":\"fe80::4a7:7dff:fe3f:5457\"},{\"name\":\"lo\",
\"macAddress\":\"00:00:00:00:00:00\",\"ipAddress\":\"::1\"}]",
                "server.performance.avgNetworkReadsPerSecondInKB":
 "0.04915364583333333",
                "server.tags": "[]",
                "server.applications.hasMoreValues": "false",
                "server.timeOfCreation": "2016-10-28 23:44:00.0",
                "server.agentId": "i-4447bc1b",
                "server.performance.maxDiskWritesPerSecondInKB": "0.0",
                "server.performance.avgDiskReadIOPS": "0.0",
                "server.performance.avgFreeRAMInKB": "1547210.1333333333",
                "server.performance.avgDiskReadsPerSecondInKB": "0.0",
                "server.performance.avgDiskWriteIOPS": "0.0",
                "server.performance.numNetworkCards": "2",
                "server.hypervisor": "xen",
                "server.networkInterfaceInfo.hasMoreValues": "false",
                "server.performance.avgNetworkWritesPerSecondInKB": "0.1380859375",
```

```
                "server.osName": "Linux - Amazon Linux AMI release 2015.03",
                "server.performance.totalRAMInKB": "1694732.0",
                "server.cpuType": "x64"
        },
        {
                "server.performance.maxCpuUsagePct": "100.0",
                "server.performance.maxDiskReadIOPS": "0.0",
                "server.performance.avgCpuUsagePct": "14.733333333333338",
                "server.type": "EC2",
                "server.performance.maxNetworkReadsPerSecondInKB": "13.400390625",
                "server.hostName": "ip-172-31-42-208",
                "server.configurationId": "d-server-099385097ef9fbcfb",
                "server.tags.hasMoreValues": "false",
                "server.performance.minFreeRAMInKB": "1531104.0",
                "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
                "server.performance.maxDiskReadsPerSecondInKB": "0.0",
                "server.applications": "[]",
                "server.performance.numDisks": "1",
                "server.performance.numCpus": "1",
                "server.performance.numCores": "1",
                "server.performance.maxDiskWriteIOPS": "1.0",
                "server.performance.maxNetworkWritesPerSecondInKB": "12.271484375",
                "server.performance.avgDiskWritesPerSecondInKB":
 "0.5333333333333334",
                "server.networkInterfaceInfo": "[{\"name\":\"eth0\",
\"macAddress\":\"06:4A:79:60:75:61\",\"ipAddress\":\"172.31.42.208\",\"netMask
\":\"255.255.240.0\"},{\"name\":\"eth0\",\"macAddress\":\"06:4A:79:60:75:61\",
\"ipAddress\":\"fe80::44a:79ff:fe60:7561\"},{\"name\":\"lo\",\"macAddress\":
\"00:00:00:00:00:00\",\"ipAddress\":\"::1\"},{\"name\":\"lo\",\"macAddress\":
\"00:00:00:00:00:00\",\"ipAddress\":\"127.0.0.1\",\"netMask\":\"255.0.0.0\"}]",
                "server.performance.avgNetworkReadsPerSecondInKB":
 "2.8720052083333334",
                "server.tags": "[]",
                "server.applications.hasMoreValues": "false",
                "server.timeOfCreation": "2016-10-28 23:44:30.0",
                "server.agentId": "i-c142b99e",
                "server.performance.maxDiskWritesPerSecondInKB": "4.0",
                "server.performance.avgDiskReadIOPS": "0.0",
                "server.performance.avgFreeRAMInKB": "1534946.4",
                "server.performance.avgDiskReadsPerSecondInKB": "0.0",
                "server.performance.avgDiskWriteIOPS": "0.13333333333333336",
                "server.performance.numNetworkCards": "2",
                "server.hypervisor": "xen",
                "server.networkInterfaceInfo.hasMoreValues": "false",
```

```
                "server.performance.avgNetworkWritesPerSecondInKB":
 "1.7977864583333332",
                "server.osName": "Linux - Amazon Linux AMI release 2015.03",
                "server.performance.totalRAMInKB": "1694732.0",
                "server.cpuType": "x64"
        }
    ]
}
```

Décrire les configurations d'actifs sélectionnées

Cet exemple de commande décrit les configurations de deux applications spécifiées. L'action détecte le type de ressource à partir de l'ID de configuration. Un seul type de ressource est autorisé par commande.

Commande :

```
aws discovery describe-configurations --configuration-ids "d-
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"
```

Sortie :

```
{
    "configurations": [
        {
                "application.serverCount": "0",
                "application.name": "Application-12345",
                "application.lastModifiedTime": "2016-12-13 23:53:27.0",
                "application.description": "",
                "application.timeOfCreation": "2016-12-13 23:53:27.0",
                "application.configurationId": "d-application-0ac39bc0e4fad0e42"
        },
        {
                "application.serverCount": "0",
                "application.name": "Application-67890",
                "application.lastModifiedTime": "2016-12-13 23:53:33.0",
                "application.description": "",
                "application.timeOfCreation": "2016-12-13 23:53:33.0",
                "application.configurationId": "d-application-02444a45288013764"
         }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConfigurationsà la section Référence des AWS CLI commandes.

`list-configurations`

L'exemple de code suivant montre comment utiliser`list-configurations`.

AWS CLI

Pour répertorier tous les serveurs découverts répondant à un ensemble de conditions de filtre

Cet exemple de commande répertorie les serveurs découverts correspondant à l'un des deux modèles de nom d'hôte et n'exécutant pas Ubuntu.

Commande :

```
aws discovery list-configurations --configuration-type SERVER --filters
 name="server.hostName",values="172-31-35","172-31-42",condition="CONTAINS"
 name="server.osName",values="Ubuntu",condition="NOT_CONTAINS"
```

Sortie :

```
{
    "configurations": [
      {
            "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
            "server.type": "EC2",
            "server.hostName": "ip-172-31-42-208",
            "server.timeOfCreation": "2016-10-28 23:44:30.0",
            "server.configurationId": "d-server-099385097ef9fbcfb",
            "server.osName": "Linux - Amazon Linux AMI release 2015.03",
            "server.agentId": "i-c142b99e"
        },
        {
            "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
            "server.type": "EC2",
            "server.hostName": "ip-172-31-35-152",
            "server.timeOfCreation": "2016-10-28 23:44:00.0",
            "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
            "server.osName": "Linux - Amazon Linux AMI release 2015.03",
            "server.agentId": "i-4447bc1b"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListConfigurationsà la section Référence des AWS CLI commandes.

Exemples de signaux d'application utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide des signaux d'application.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-get-service-level-objective-budget-report`

L'exemple de code suivant montre comment utiliser`batch-get-service-level-objective-budget-report`.

AWS CLI

Pour récupérer un ou plusieurs rapports budgétaires sur les objectifs de niveau de service (SLO).

L'`batch-get-service-level-objective-budget-report`exemple suivant extrait un ou plusieurs rapports budgétaires sur les objectifs de niveau de service (SLO).

```
aws application-signals batch-get-service-level-objective-budget-report \
--timestamp 1735059869 \
--slo-ids "arn:aws:application-signals:us-east-1:123456789101:slo/
SLOName1" "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName2"
```

Sortie :

```
{
    "Timestamp": "2024-12-24T22:34:29+05:30",
    "Reports": [{
            "Arn": "arn:aws:application-signals:us-east-1:123456789101:slo/
SLOName1",
            "Name": "Your SLO Name",
            "EvaluationType": "PeriodBased",
            "BudgetStatus": "OK",
            "Attainment": 100.0,
            "TotalBudgetSeconds": 6048,
            "BudgetSecondsRemaining": 6048,
            "Sli": {
                "SliMetric": {
                    "MetricDataQueries": [{
                        "Id": "m1",
                        "MetricStat": {
                            "Metric": {
                                "Namespace": "AWS/EC2",
                                "MetricName": "CPUUtilization",
                                "Dimensions": [{
                                    "Name": "InstanceId",
                                    "Value": "i-0e098765432522"
                                }]
                            },
                            "Period": 60,
                            "Stat": "Average"
                        },
                        "ReturnData": true
                    }]
                },
                "MetricThreshold": 200.0,
                "ComparisonOperator": "LessThanOrEqualTo"
            },
            "Goal": {
                "Interval": {
                    "RollingInterval": {
                        "DurationUnit": "DAY",
                        "Duration": 7
                    }
                },
                "AttainmentGoal": 99.0,
                "WarningThreshold": 50.0
```

```
            }
        },
        {
            "Arn": "arn:aws:application-signals:us-east-1:123456789101:slo/
SLOName2",
            "Name": "test",
            "EvaluationType": "PeriodBased",
            "BudgetStatus": "BREACHED",
            "Attainment": 97.39583275,
            "TotalBudgetSeconds": 86,
            "BudgetSecondsRemaining": -2154,
            "Sli": {
                "SliMetric": {
                    "MetricDataQueries": [{
                        "Id": "cwMetric",
                        "MetricStat": {
                            "Metric": {
                                "Namespace": "AWS/EC2",
                                "MetricName": "CPUUtilization",
                                "Dimensions": [{
                                    "Name": "InstanceId",
                                    "Value": "i-0e12345678922"
                                }]
                            },
                            "Period": 300,
                            "Stat": "Average"
                        },
                        "ReturnData": true
                    }]
                },
                "MetricThreshold": 5.0,
                "ComparisonOperator": "GreaterThan"
            },
            "Goal": {
                "Interval": {
                    "RollingInterval": {
                        "DurationUnit": "DAY",
                        "Duration": 1
                    }
                },
                "AttainmentGoal": 99.9,
                "WarningThreshold": 30.0
            }
        }
```

```
    ],
    "Errors": []
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir BatchGetServiceLevelObjectiveBudgetReportla section Référence des AWS CLI commandes.

`create-service-level-objective`

L'exemple de code suivant montre comment utiliser`create-service-level-objective`.

AWS CLI

Pour créer un objectif de niveau de service (SLO)

L'`create-service-level-objective`exemple suivant crée un objectif de niveau de service (SLO), qui peut vous aider à garantir que vos opérations commerciales critiques répondent aux attentes des clients.

```
aws application-signals create-service-level-objective \
    --name "SLOName" \
    --description "Description of your SLO" \
    --sli-config file://sli-config.json
```

Contenu de `sli-config.json` :

```
{
    "SliMetricConfig": {
        "MetricDataQueries": [
            {
                "Id": "m1",
                "MetricStat": {
                    "Metric": {
                        "Namespace": "AWS/EC2",
                        "MetricName": "CPUUtilization",
                        "Dimensions": [
                            {
                                "Name": "InstanceId",
                                "Value": "i-0e5a1234561522"
```

```
                            }
                        ]
                    },
                    "Period": 60,
                    "Stat": "Average"
                },
                "ReturnData": true
            }
        ]
    },
    "MetricThreshold": 200,
    "ComparisonOperator": "LessThanOrEqualTo"
}
```

Sortie :

```
{
    "Slo": {
    "Arn": "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName",
    "Name": "SLOName",
    "Description": "Description of your SLO",
    "CreatedTime": "2024-12-27T08:16:09.032000+05:30",
    "LastUpdatedTime": "2024-12-27T08:16:09.032000+05:30",
    "Sli": {
        "SliMetric": {
            "MetricDataQueries": [
                {
                    "Id": "m1",
                    "MetricStat": {
                        "Metric": {
                            "Namespace": "AWS/EC2",
                            "MetricName": "CPUUtilization",
                            "Dimensions": [
                                {
                                    "Name": "InstanceId",
                                    "Value": "i-0e59876543234522"
                                }
                            ]
                        },
                        "Period": 60,
                        "Stat": "Average"
                    },
                    "ReturnData": true
```

```
                }
            ]
        },
        "MetricThreshold": 200.0,
        "ComparisonOperator": "LessThanOrEqualTo"
    },
    "EvaluationType": "PeriodBased",
    "Goal": {
        "Interval": {
            "RollingInterval": {
                "DurationUnit": "DAY",
                "Duration": 7
            }
        },
        "AttainmentGoal": 99.0,
        "WarningThreshold": 50.0
      }
    }
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir CreateServiceLevelObjectivela section Référence des AWS CLI commandes.

`delete-service-level-objective`

L'exemple de code suivant montre comment utiliser`delete-service-level-objective`.

AWS CLI

Pour supprimer l'objectif de niveau de service spécifié.

L'`delete-service-level-objective`exemple suivant supprime l'objectif de niveau de service spécifié.

```
aws application-signals delete-service-level-objective \
    --id "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir DeleteServiceLevelObjectivela section Référence des AWS CLI commandes.

`get-service-level-objective`

L'exemple de code suivant montre comment utiliser`get-service-level-objective`.

AWS CLI

Pour renvoyer des informations sur un SLO créé dans le compte

L'`get-service-level-objective`exemple suivant renvoie des informations sur un SLO créé dans le compte.

```
aws application-signals get-service-level-objective \
    --id "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName"
```

Sortie :

```
{
    "Slo": {
        "Arn": "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName",
        "Name": "SLOName",
        "Description": "Description of your SLO",
        "CreatedTime": "2024-12-24T22:19:18.624000+05:30",
        "LastUpdatedTime": "2024-12-24T22:19:55.280000+05:30",
        "Sli": {
            "SliMetric": {
                "MetricDataQueries": [{
                    "Id": "m1",
                    "MetricStat": {
                        "Metric": {
                            "Namespace": "AWS/EC2",
                            "MetricName": "CPUUtilization",
                            "Dimensions": [{
                                "Name": "InstanceId",
                                "Value": "i-0e0987654321522"
                            }]
                        },
```

```
                        "Period": 60,
                        "Stat": "Average"
                    },
                    "ReturnData": true
                }]
            },
            "MetricThreshold": 200.0,
            "ComparisonOperator": "LessThanOrEqualTo"
        },
        "EvaluationType": "PeriodBased",
        "Goal": {
            "Interval": {
                "RollingInterval": {
                    "DurationUnit": "DAY",
                    "Duration": 7
                }
            },
            "AttainmentGoal": 99.0,
            "WarningThreshold": 50.0
        }
    }
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir GetServiceLevelObjectivela section Référence des AWS CLI commandes.

`get-service`

L'exemple de code suivant montre comment utiliser`get-service`.

AWS CLI

Pour renvoyer des informations sur un service découvert par Application Signals

L'`get-service`exemple suivant renvoie des informations sur un service découvert par Application Signals.

```
aws application-signals get-service \
    --start-time 1732704000 \
    --end-time 1732714500 \
```

```
    --key-attributes Environment=lambda:default,Name=hello-world-python,Type=Service
```

Sortie :

```
{
    "Service": {
        "KeyAttributes": {
            "Environment": "lambda:default",
            "Name": "hello-world-python",
            "Type": "Service"
        },
        "AttributeMaps": [{
            "Lambda.Function.Name": "hello-world-python",
            "PlatformType": "AWS::Lambda"
        }],
        "MetricReferences": [{
            "Namespace": "ApplicationSignals",
            "MetricType": "LATENCY",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "lambda:default"
            }, {
                "Name": "Service",
                "Value": "hello-world-python"
            }],
            "MetricName": "Latency"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "FAULT",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "lambda:default"
            }, {
                "Name": "Service",
                "Value": "hello-world-python"
            }],
            "MetricName": "Fault"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "ERROR",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "lambda:default"
```

```
            }, {
                "Name": "Service",
                "Value": "hello-world-python"
            }],
            "MetricName": "Error"
        }],
        "LogGroupReferences": [{
            "Identifier": "/aws/lambda/hello-world-python",
            "ResourceType": "AWS::Logs::LogGroup",
            "Type": "AWS::Resource"
        }]
    },
    "StartTime": "2024-11-27T10:00:00+00:00",
    "EndTime": "2024-11-27T14:00:01+00:00",
    "LogGroupReferences": [{
        "Identifier": "/aws/lambda/hello-world-python",
        "ResourceType": "AWS::Logs::LogGroup",
        "Type": "AWS::Resource"
    }]
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir GetServicela section Référence des AWS CLI commandes.

`list-service-dependencies`

L'exemple de code suivant montre comment utiliser`list-service-dependencies`.

AWS CLI

Pour renvoyer une liste des dépendances du service que vous spécifiez

L'`list-service-dependencies`exemple suivant renvoie une liste des dépendances du service que vous spécifiez.

```
aws application-signals list-service-dependencies \
    --start-time 1732021200 \
    --end-time 1732107600 \
    --key-attributes Environment=api-
gateway:prod, Name=PetAdoptionStatusUpdater,Type=Service
```

Sortie :


```
{
    "ServiceDependencies": [{
        "OperationName": "PUT /prod",
        "DependencyKeyAttributes": {
            "Environment": "lambda:default",
            "Name": "Services-name",
            "Type": "Service"
        },
        "DependencyOperationName": "Invoke",
        "MetricReferences": [{
            "Namespace": "ApplicationSignals",
            "MetricType": "LATENCY",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "api-gateway:prod"
            }, {
                "Name": "Operation",
                "Value": "PUT /prod"
            }, {
                "Name": "RemoteEnvironment",
                "Value": "lambda:default"
            }, {
                "Name": "RemoteOperation",
                "Value": "Invoke"
            }, {
                "Name": "RemoteService",
                "Value": "Services-name"
            }, {
                "Name": "Service",
                "Value": "PetAdoptionStatusUpdater"
            }],
            "MetricName": "Latency"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "FAULT",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "api-gateway:prod"
            }, {
                "Name": "Operation",
                "Value": "PUT /prod"
            }, {
```

```
                "Name": "RemoteEnvironment",
                "Value": "lambda:default"
            }, {
                "Name": "RemoteOperation",
                "Value": "Invoke"
            }, {
                "Name": "RemoteService",
                "Value": "Services-name"
            }, {
                "Name": "Service",
                "Value": "PetAdoptionStatusUpdater"
            }],
            "MetricName": "Fault"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "ERROR",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "api-gateway:prod"
            }, {
                "Name": "Operation",
                "Value": "PUT /prod"
            }, {
                "Name": "RemoteEnvironment",
                "Value": "lambda:default"
            }, {
                "Name": "RemoteOperation",
                "Value": "Invoke"
            }, {
                "Name": "RemoteService",
                "Value": "Services-name"
            }, {
                "Name": "Service",
                "Value": "PetAdoptionStatusUpdater"
            }],
            "MetricName": "Error"
        }]
    }],
    "StartTime": "2024-11-19T13:00:00+00:00",
    "EndTime": "2024-11-20T13:00:01+00:00"
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListServiceDependencieslа section Référence des AWS CLI commandes.

`list-service-dependents`

L'exemple de code suivant montre comment utiliser`list-service-dependents`.

AWS CLI

Pour renvoyer la liste des personnes à charge qui ont invoqué le service spécifié pendant la période spécifiée

L'`list-service-dependents`exemple suivant renvoie la liste des personnes à charge qui ont invoqué le service spécifié au cours de la période spécifiée.

```
aws application-signals list-service-dependents \
    --start-time 1732021200 \
    --end-time 1732107600 \
    --key-attributes Environment=generic:default,Name=PetSite,Type=Service
```

Sortie :

```
{
    "ServiceDependents": [{
        "OperationName": "",
        "DependentKeyAttributes": {
            "Identifier": "pet-api-canary-hao",
            "ResourceType": "AWS::Synthetics::Canary",
            "Type": "AWS::Resource"
        },
        "DependentOperationName": "",
        "MetricReferences": []
    }, {
        "OperationName": "",
        "DependentKeyAttributes": {
            "Identifier": "PetSite",
            "ResourceType": "AWS::Synthetics::Canary",
            "Type": "AWS::Resource"
        },
```

```
        "DependentOperationName": "",
        "MetricReferences": []
    }],
    "StartTime": "2024-12-24T05:00:00+00:00",
    "EndTime": "2024-12-25T06:00:01+00:00"
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListServiceDependentsla section Référence des AWS CLI commandes.

`list-service-level-objectives`

L'exemple de code suivant montre comment utiliser`list-service-level-objectives`.

AWS CLI

Pour renvoyer une liste des personnes SLOs créées dans ce compte.

L'`list-service-level-objectives`exemple suivant renvoie une liste des comptes SLOs créés dans ce compte.

```
aws application-signals list-service-level-objectives
```

Sortie :

```
{
    "SloSummaries": [{
        "Arn": "arn:aws:application-signals:us-east-1:123456789101:slo/test",
        "Name": "test",
        "CreatedTime": "2024-12-24T22:01:21.116000+05:30"
    }]
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListServiceLevelObjectivesla section Référence des AWS CLI commandes.

`list-service-operations`

L'exemple de code suivant montre comment utiliser`list-service-operations`.

AWS CLI

Pour renvoyer une liste des opérations de ce service qui ont été découvertes par Application Signals

L'`list-service-operations`exemple suivant renvoie une liste des opérations de ce service qui ont été découvertes par Application Signals.

```
aws application-signals list-service-operations \
    --start-time 1735017423 \
    --end-time 1735103823 \
    --key-attributes Environment=generic:default,Name=payforadoption,Type=Service
```

Sortie :

```
{
    "ServiceOperations": [{
        "Name": "POST /api",
        "MetricReferences": [{
            "Namespace": "ApplicationSignals",
            "MetricType": "LATENCY",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "generic:default"
            }, {
                "Name": "Operation",
                "Value": "POST /api"
            }, {
                "Name": "Service",
                "Value": "payforadoption"
            }],
            "MetricName": "Latency"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "FAULT",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "generic:default"
            }, {
```

```
                "Name": "Operation",
                "Value": "POST /api"
            }, {
                "Name": "Service",
                "Value": "payforadoption"
            }],
            "MetricName": "Fault"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "ERROR",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "generic:default"
            }, {
                "Name": "Operation",
                "Value": "POST /api"
            }, {
                "Name": "Service",
                "Value": "payforadoption"
            }],
            "MetricName": "Error"
        }]
    }],
    "StartTime": "2024-12-24T05:00:00+00:00",
    "EndTime": "2024-12-25T06:00:01+00:00"
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListServiceOperationsla section Référence des AWS CLI commandes.

`list-services`

L'exemple de code suivant montre comment utiliser`list-services`.

AWS CLI

Pour renvoyer une liste des services découverts par Application Signals

L'`list-services`exemple suivant renvoie une liste des services découverts par Application Signals.

```
aws application-signals list-services \
    --start-time 1734918791 \
    --end-time 1734965591
```

Sortie :


```
{
    "ServiceSummaries": [{
        "KeyAttributes": {
            "Environment": "lambda:default",
            "Name": "hello-world-python",
            "Type": "Service"
        },
        "AttributeMaps": [{
            "Lambda.Function.Name": "hello-world-python",
            "PlatformType": "AWS::Lambda"
        }],
        "MetricReferences": [{
            "Namespace": "ApplicationSignals",
            "MetricType": "LATENCY",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "lambda:default"
            }, {
                "Name": "Service",
                "Value": "hello-world-python"
            }],
            "MetricName": "Latency"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "FAULT",
            "Dimensions": [{
                "Name": "Environment",
                "Value": "lambda:default"
            }, {
                "Name": "Service",
                "Value": "hello-world-python"
            }],
            "MetricName": "Fault"
        }, {
            "Namespace": "ApplicationSignals",
            "MetricType": "ERROR",
            "Dimensions": [{
```

```
                "Name": "Environment",
                "Value": "lambda:default"
            }, {
                "Name": "Service",
                "Value": "hello-world-python"
            }],
            "MetricName": "Error"
        }]
    }],
    "StartTime": "2024-11-27T10:00:00+00:00",
    "EndTime": "2024-11-27T14:00:01+00:00"
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListServicesla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour afficher les balises associées à une CloudWatch ressource

L'`list-tags-for-resource`exemple suivant affiche les balises associées à une CloudWatch ressource.

```
aws application-signals list-tags-for-resource \
    --resource-arn "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName"
```

Sortie :

```
{
    "Tags": [{
        "Key": "test",
        "Value": "value"
    }]
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`start-discovery`

L'exemple de code suivant montre comment utiliser`start-discovery`.

AWS CLI

Pour permettre à ce compte Amazon Web Services d'utiliser les signaux CloudWatch d'application

L'`start-discovery`exemple suivant permet à ce compte Amazon Web Services d'utiliser les signaux d' CloudWatch application en créant le rôle AWS ServiceRoleForCloudWatchApplicationSignalslié au service.

```
aws application-signals start-discovery
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir StartDiscoveryla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour attribuer une ou plusieurs balises (paires clé-valeur) à la CloudWatch ressource spécifiée, telle qu'un objectif de niveau de service

L'`tag-resource`exemple suivant attribue une ou plusieurs balises (paires clé-valeur) à la CloudWatch ressource spécifiée, par exemple un objectif de niveau de service.

```
aws application-signals tag-resource \
```

```
    --resource-arn "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName"
\
    --tags '{"Key":"test","Value":"value"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une ou plusieurs balises de la ressource spécifiée

L'`untag-resource`exemple suivant supprime une ou plusieurs balises de la ressource spécifiée.

```
aws application-signals untag-resource \
    --resource-arn "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName"
 \
    --tag-keys "test"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-service-level-objective`

L'exemple de code suivant montre comment utiliser`update-service-level-objective`.

AWS CLI

Pour mettre à jour un objectif de niveau de service (SLO) existant

L'`update-service-level-objective`exemple suivant met à jour un objectif de niveau de service (SLO) existant.

```
aws application-signals update-service-level-objective \
--cli-input-json file://update-slo.json
```

Contenu de `update-slo.json` :

```
{
    "id": "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName",
    "goal": {
        "Interval": {
            "RollingInterval": {
                "DurationUnit": "DAY",
                "Duration": 7
            }
        },
        "AttainmentGoal": 90.0,
        "WarningThreshold": 50.0
    }
}
```

Sortie :

```
{
    "Slo": {
        "Arn": "arn:aws:application-signals:us-east-1:123456789101:slo/SLOName",
        "Name": "SLOName",
        "Description": "Description of your SLO",
        "CreatedTime": "2024-12-24T22:19:18.624000+05:30",
        "LastUpdatedTime": "2024-12-27T08:51:38.278000+05:30",
        "Sli": {
            "SliMetric": {
                "MetricDataQueries": [{
                    "Id": "m1",
                    "MetricStat": {
                        "Metric": {
                            "Namespace": "AWS/EC2",
                            "MetricName": "CPUUtilization",
                            "Dimensions": [{
                                "Name": "InstanceId",
                                "Value": "i-00987654345222"
```

```
                            }]
                        },
                        "Period": 60,
                        "Stat": "Average"
                    },
                    "ReturnData": true
                }]
            },
            "MetricThreshold": 200.0,
            "ComparisonOperator": "LessThanOrEqualTo"
        },
        "EvaluationType": "PeriodBased",
        "Goal": {
            "Interval": {
                "RollingInterval": {
                    "DurationUnit": "DAY",
                    "Duration": 7
                }
            },
            "AttainmentGoal": 90.0,
            "WarningThreshold": 50.0
        }
    }
}
```

Pour plus d'informations, consultez Application Signals dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, voir UpdateServiceLevelObjectivela section Référence des AWS CLI commandes.

AppRegistry exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AppRegistry.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-attribute-group`

L'exemple de code suivant montre comment utiliser`associate-attribute-group`.

AWS CLI

Pour associer un groupe d'attributs

L'`associate-attribute-group`exemple suivant associe un groupe d'attributs spécifique de votre AWS compte à une application spécifique de votre AWS compte.

```
aws servicecatalog-appregistry associate-attribute-group \
    --application "ExampleApplication" \
    --attribute-group "ExampleAttributeGroup"
```

Sortie :

```
{
    "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
    "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl"
}
```

Pour plus d'informations, consultez la section Associer et dissocier des groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous AssociateAttributeGroupà la section Référence des AWS CLI commandes.

`create-application`

L'exemple de code suivant montre comment utiliser`create-application`.

AWS CLI

Pour créer une application

L'`create-application`exemple suivant crée une nouvelle application dans votre AWS compte.

```
aws servicecatalog-appregistry create-application \
    --name "ExampleApplication"
```

Sortie :

```
{
    "application": {
        "id": "0ars38r6btoohvpvd9gqrptt9l",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
        "name": "ExampleApplication",
        "creationTime": "2023-02-28T21:10:10.820000+00:00",
        "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Création d'applications dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous CreateApplicationà la section Référence des AWS CLI commandes.

`create-attribute-group`

L'exemple de code suivant montre comment utiliser`create-attribute-group`.

AWS CLI

Pour créer un groupe d'attributs

L'`create-attribute-group`exemple suivant crée un nouveau groupe d'attributs dans votre AWS compte.

```
aws servicecatalog-appregistry create-attribute-group \
```

```
    --name "ExampleAttributeGroup" \
    --attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'
```

Sortie :

```
{
    "attributeGroup": {
        "id": "01sj5xdwhbw54kejwnt09fnpcl",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
        "name": "ExampleAttributeGroup",
        "creationTime": "2023-02-28T20:38:01.389000+00:00",
        "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Création de groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous CreateAttributeGroupà la section Référence des AWS CLI commandes.

`delete-application`

L'exemple de code suivant montre comment utiliser`delete-application`.

AWS CLI

Pour supprimer une application

L'`delete-application`exemple suivant supprime une application spécifique de votre AWS compte.

```
aws servicecatalog-appregistry delete-application \
    --application "ExampleApplication3"
```

Sortie :

```
{
```

```
    "application": {
        "id": "055gw7aynr1i5mbv7kjwzx5945",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
        "name": "ExampleApplication3",
        "creationTime": "2023-02-28T22:06:28.228000+00:00",
        "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Suppression d'applications dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous DeleteApplicationà la section Référence des AWS CLI commandes.

`delete-attribute-group`

L'exemple de code suivant montre comment utiliser`delete-attribute-group`.

AWS CLI

Exemple 8 : Pour supprimer un groupe d'attributs

L'`delete-attribute-group`exemple suivant supprime un groupe d'attributs spécifique de votre AWS compte.

```
aws servicecatalog-appregistry delete-attribute-group \
    --attribute-group "ExampleAttributeGroup3"
```

Sortie :

```
{
    "attributeGroup": {
        "id": "011ge6y3emyjijt8dw8jn6r0hv",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
        "name": "ExampleAttributeGroup3",
        "creationTime": "2023-02-28T22:05:35.224000+00:00",
        "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
    }
```

```
}
```

Pour plus d'informations, consultez la section Suppression de groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous DeleteAttributeGroupà la section Référence des AWS CLI commandes.

`get-application`

L'exemple de code suivant montre comment utiliser`get-application`.

AWS CLI

Pour obtenir une candidature

L'`get-application`exemple suivant permet de récupérer les informations de métadonnées relatives à une application spécifique de votre AWS compte.

```
aws servicecatalog-appregistry get-application \
    --application "ExampleApplication"
```

Sortie :

```
{
    "id": "0ars38r6btoohvpvd9gqrptt9l",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
    "name": "ExampleApplication",
    "creationTime": "2023-02-28T21:10:10.820000+00:00",
    "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",
    "associatedResourceCount": 0,
    "tags": {
        "aws:servicecatalog:applicationName": "ExampleApplication"
    },
    "integrations": {
        "resourceGroup": {
            "state": "CREATE_COMPLETE",
            "arn": "arn:aws:resource-groups:us-west-2:813737243517:group/
AWS_AppRegistry_Application-ExampleApplication"
        }
    }
```

```
}
```

Pour plus d'informations, consultez les détails de l'utilisation de l'application dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous GetApplicationà la section Référence des AWS CLI commandes.

`get-attribute-group`

L'exemple de code suivant montre comment utiliser`get-attribute-group`.

AWS CLI

Pour obtenir un groupe d'attributs

L'`get-attribute-group`exemple suivant permet de récupérer un groupe d'attributs spécifique dans votre AWS compte.

```
aws servicecatalog-appregistry get-attribute-group \
    --attribute-group "ExampleAttributeGroup"
```

Sortie :

```
{
    "id": "01sj5xdwhbw54kejwnt09fnpcl",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
    "name": "ExampleAttributeGroup",
    "attributes": "{\"SomeKey1\":\"SomeValue1\",\"SomeKey2\":\"SomeValue2\"}",
    "creationTime": "2023-02-28T20:38:01.389000+00:00",
    "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",
    "tags": {
        "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"
    }
}
```

Pour plus d'informations, consultez la section Gestion des métadonnées pour les groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous GetAttributeGroupà la section Référence des AWS CLI commandes.

`list-applications`

L'exemple de code suivant montre comment utiliser`list-applications`.

AWS CLI

Pour répertorier les applications

L'`list-applications`exemple suivant permet de récupérer la liste de toutes les applications de votre AWS compte.

```
aws servicecatalog-appregistry list-applications
```

Sortie :

```
{
    "applications": [
        {
            "id": "03axw94pjfj3uan00tcgbrxnkw",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/03axw94pjfj3uan00tcgbrxnkw",
            "name": "ExampleApplication2",
            "creationTime": "2023-02-28T21:59:34.094000+00:00",
            "lastUpdateTime": "2023-02-28T21:59:34.094000+00:00"
        },
        {
            "id": "055gw7aynr1i5mbv7kjwzx5945",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
            "name": "ExampleApplication3",
            "creationTime": "2023-02-28T22:06:28.228000+00:00",
            "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
        },
        {
            "id": "0ars38r6btoohvpvd9gqrptt9l",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
            "name": "ExampleApplication",
            "description": "This is an example application",
            "creationTime": "2023-02-28T21:10:10.820000+00:00",
            "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Affichage des détails de l'application dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous ListApplicationsà la section Référence des AWS CLI commandes.

`list-associated-attribute-groups`

L'exemple de code suivant montre comment utiliser`list-associated-attribute-groups`.

AWS CLI

Pour répertorier les groupes d'attributs associés

L'`list-associated-attribute-groups`exemple suivant extrait une liste de tous les groupes d'attributs de votre AWS compte associés à une application spécifique de votre AWS compte.

```
aws servicecatalog-appregistry list-associated-attribute-groups \
    --application "ExampleApplication"
```

Sortie :

```
{
    "attributeGroups": [
        "01sj5xdwhbw54kejwnt09fnpcl"
    ]
}
```

Pour plus d'informations, consultez la section Associer et dissocier des groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous ListAssociatedAttributeGroupsà la section Référence des AWS CLI commandes.

`list-attribute-groups-for-application`

L'exemple de code suivant montre comment utiliser`list-attribute-groups-for-application`.

AWS CLI

Pour répertorier les groupes d'attributs d'une application

L'`list-attribute-groups-for-application`exemple suivant répertorie les détails de tous les groupes d'attributs de votre AWS compte associés à une application spécifique de votre AWS compte.

```
aws servicecatalog-appregistry list-attribute-groups-for-application \
    --application "ExampleApplication"
```

Sortie :

```
{
    "attributeGroupsDetails": [
        {
            "id": "01sj5xdwhbw54kejwnt09fnpcl",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
            "name": "ExampleAttributeGroup"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails des groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous ListAttributeGroupsForApplicationà la section Référence des AWS CLI commandes.

`list-attribute-groups`

L'exemple de code suivant montre comment utiliser`list-attribute-groups`.

AWS CLI

Pour répertorier les groupes d'attributs

L'`list-attribute-groups`exemple suivant permet de récupérer la liste de tous les groupes d'attributs de votre AWS compte.

```
aws servicecatalog-appregistry list-attribute-groups
```

Sortie :

```
{
    "attributeGroups": [
        {
            "id": "011ge6y3emyjijt8dw8jn6r0hv",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
            "name": "ExampleAttributeGroup3",
            "creationTime": "2023-02-28T22:05:35.224000+00:00",
            "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
        },
        {
            "id": "01sj5xdwhbw54kejwnt09fnpcl",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
            "name": "ExampleAttributeGroup",
            "description": "This is an example attribute group",
            "creationTime": "2023-02-28T20:38:01.389000+00:00",
            "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00"
        },
        {
            "id": "03n1yffgq6d18vwrzxf0c70nm3",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/03n1yffgq6d18vwrzxf0c70nm3",
            "name": "ExampleAttributeGroup2",
            "creationTime": "2023-02-28T21:57:30.687000+00:00",
            "lastUpdateTime": "2023-02-28T21:57:30.687000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails des groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous ListAttributeGroupsà la section Référence des AWS CLI commandes.

`update-application`

L'exemple de code suivant montre comment utiliser`update-application`.

AWS CLI

Pour mettre à jour une application

L'`update-application`exemple suivant met à jour une application spécifique de votre AWS compte pour y inclure une description.

```
aws servicecatalog-appregistry update-application \
    --application "ExampleApplication" \
    --description "This is an example application"
```

Sortie :

```
{
    "application": {
        "id": "0ars38r6btoohvpvd9gqrptt9l",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
        "name": "ExampleApplication",
        "description": "This is an example application",
        "creationTime": "2023-02-28T21:10:10.820000+00:00",
        "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00",
        "tags": {
            "aws:servicecatalog:applicationName": "ExampleApplication"
        }
    }
}
```

Pour plus d'informations, consultez la section Modification d'applications dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous UpdateApplicationà la section Référence des AWS CLI commandes.

`update-attribute-group`

L'exemple de code suivant montre comment utiliser`update-attribute-group`.

AWS CLI

Pour mettre à jour un groupe d'attributs

L'`update-attribute-group`exemple suivant met à jour un groupe d'attributs spécifique de votre AWS compte pour inclure une description.

```
aws servicecatalog-appregistry update-attribute-group \
    --attribute-group "ExampleAttributeGroup" \
    --description "This is an example attribute group"
```

Sortie :

```
{
    "attributeGroup": {
        "id": "01sj5xdwhbw54kejwnt09fnpcl",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
        "name": "ExampleAttributeGroup",
        "description": "This is an example attribute group",
        "creationTime": "2023-02-28T20:38:01.389000+00:00",
        "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00",
        "tags": {
            "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"
        }
    }
}
```

Pour plus d'informations, consultez la section Modification de groupes d'attributs dans le Guide de l' AppRegistry administrateur du AWS Service Catalog.

- Pour plus de détails sur l'API, reportez-vous UpdateAttributeGroupà la section Référence des AWS CLI commandes.

Exemples d'Athena utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Athena.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-get-named-query`

L'exemple de code suivant montre comment utiliser`batch-get-named-query`.

AWS CLI

Pour renvoyer des informations relatives à plusieurs requêtes

L'`batch-get-named-query`exemple suivant renvoie des informations sur les requêtes nommées qui ont la valeur spécifiée IDs.

```
aws athena batch-get-named-query \
    --named-query-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

Sortie :

```
{
    "NamedQueries": [
        {
            "Name": "Flights Select Query",
            "Description": "Sample query to get the top 10 airports with the most number of departures since 2000",
            "Database": "sampledb",
            "QueryString": "SELECT origin, count(*) AS total_departures\nFROM \nflights_parquet\nWHERE year >= '2000'\nGROUP BY origin\nORDER BY total_departures DESC\nLIMIT 10;",
            "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "WorkGroup": "primary"
        },
        {
            "Name": "Load flights table partitions",
            "Description": "Sample query to load flights table partitions using MSCK REPAIR TABLE statement",
            "Database": "sampledb",
            "QueryString": "MSCK REPAIR TABLE flights_parquet;",
            "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
            "WorkGroup": "primary"
        },
        {
            "Name": "CloudFront Select Query",
            "Description": "Sample query to view requests per operating system
 during a particular time frame",
            "Database": "sampledb",
            "QueryString": "SELECT os, COUNT(*) count FROM cloudfront_logs WHERE
 date BETWEEN date '2014-07-05' AND date '2014-08-05' GROUP BY os;",
            "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "WorkGroup": "primary"
        }
    ],
    "UnprocessedNamedQueryIds": []
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous BatchGetNamedQueryà la section Référence des AWS CLI commandes.

`batch-get-query-execution`

L'exemple de code suivant montre comment utiliser`batch-get-query-execution`.

AWS CLI

Pour renvoyer des informations sur une ou plusieurs exécutions de requêtes

L'`batch-get-query-execution`exemple suivant renvoie des informations d'exécution pour les requêtes contenant la requête spécifiée IDs.

```
aws athena batch-get-query-execution \
    --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222
```

Sortie :

```
{
    "QueryExecutions": [
        {
            "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
            "Query": "create database if not exists webdata",
            "StatementType": "DDL",
            "ResultConfiguration": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111.txt"
            },
            "QueryExecutionContext": {},
            "Status": {
                "State": "SUCCEEDED",
                "SubmissionDateTime": 1593470720.592,
                "CompletionDateTime": 1593470720.902
            },
            "Statistics": {
                "EngineExecutionTimeInMillis": 232,
                "DataScannedInBytes": 0,
                "TotalExecutionTimeInMillis": 310,
            "ResultConfiguration": {

                "QueryQueueTimeInMillis": 50,
                "ServiceProcessingTimeInMillis": 28
            },
            "WorkGroup": "AthenaAdmin"
        },
        {
            "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Query": "select date, location, browser, uri, status from
 cloudfront_logs where method = 'GET' and status = 200 and location like 'SFO%'
 limit 10",
            "StatementType": "DML",
            "ResultConfiguration": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222.csv"
            },
            "QueryExecutionContext": {
                "Database": "mydatabase",
                "Catalog": "awsdatacatalog"
            },
            "Status": {
                "State": "SUCCEEDED",
                "SubmissionDateTime": 1593469842.665,
                "CompletionDateTime": 1593469846.486
            },
            "Statistics": {
                "EngineExecutionTimeInMillis": 3600,
```

```
                "DataScannedInBytes": 203089,
                "TotalExecutionTimeInMillis": 3821,
                "QueryQueueTimeInMillis": 267,
                "QueryPlanningTimeInMillis": 1175
            },
            "WorkGroup": "AthenaAdmin"
        }
    ],
    "UnprocessedQueryExecutionIds": []
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous BatchGetQueryExecutionà la section Référence des AWS CLI commandes.

`create-data-catalog`

L'exemple de code suivant montre comment utiliser`create-data-catalog`.

AWS CLI

Pour créer un catalogue de données

L'`create-data-catalog`exemple suivant crée le catalogue `dynamo_db_catalog` de données.

```
aws athena create-data-catalog \
    --name dynamo_db_catalog \
    --type LAMBDA \
    --description "DynamoDB Catalog" \
    --parameters function=arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda
```

Cette commande ne produit aucun résultat. Pour voir le résultat, utilisez`aws athena get-data-catalog --name dynamo_db_catalog`.

Pour plus d'informations, consultez la section Enregistrement d'un catalogue : create-data-catalog dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous CreateDataCatalogà la section Référence des AWS CLI commandes.

`create-named-query`

L'exemple de code suivant montre comment utiliser`create-named-query`.

AWS CLI

Pour créer une requête nommée

L'`create-named-query`exemple suivant crée une requête enregistrée dans le `AthenaAdmin` groupe de travail qui interroge le `flights_parquet` tableau pour les vols entre Seattle et New York en janvier 2016 dont le départ et l'arrivée ont tous deux été retardés de plus de dix minutes. Les valeurs du code d'aéroport figurant dans le tableau étant des chaînes comportant des guillemets doubles (par exemple, « SEA »), elles sont protégées par des barres obliques inverses et entourées de guillemets simples.

```
aws athena create-named-query \
    --name "SEA to JFK delayed flights Jan 2016" \
    --description "Both arrival and departure delayed more than 10 minutes." \
    --database sampledb \
    --query-string "SELECT flightdate, carrier, flightnum, origin, dest,
 depdelayminutes, arrdelayminutes FROM sampledb.flights_parquet WHERE yr = 2016 AND
 month = 1 AND origin = '\"SEA\"' AND dest = '\"JFK\"' AND depdelayminutes > 10 AND
 arrdelayminutes > 10" \
    --work-group AthenaAdmin
```

Sortie :

```
{
    "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous CreateNamedQueryà la section Référence des AWS CLI commandes.

`create-work-group`

L'exemple de code suivant montre comment utiliser`create-work-group`.

AWS CLI

Pour créer un groupe de travail

L'`create-work-group`exemple suivant crée un groupe de travail appelé `Data_Analyst_Group` qui possède l'emplacement `s3://amzn-s3-demo-bucket` de sortie des résultats de la requête. La commande crée un groupe de travail qui remplace les paramètres de configuration du client, y compris l'emplacement de sortie des résultats de la requête. La commande active également CloudWatch les métriques et ajoute trois paires de balises clé-valeur au groupe de travail pour le distinguer des autres groupes de travail. Notez que l'`--configuration`argument ne comporte aucun espace avant les virgules qui séparent ses options.

```
aws athena create-work-group \
    --name Data_Analyst_Group \
    --configuration ResultConfiguration={OutputLocation="s3://amzn-s3-demo-
bucket"},EnforceWorkGroupConfiguration="true",PublishCloudWatchMetricsEnabled="true"
 \
    --description "Workgroup for data analysts" \
    --tags Key=Division,Value=West Key=Location,Value=Seattle Key=Team,Value="Big
 Data"
```

Cette commande ne produit aucun résultat. Pour voir les résultats, utilisez`aws athena get-work-group --work-group Data_Analyst_Group`.

Pour plus d'informations, consultez la section Gestion des groupes de travail dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous CreateWorkGroupà la section Référence des AWS CLI commandes.

`delete-data-catalog`

L'exemple de code suivant montre comment utiliser`delete-data-catalog`.

AWS CLI

Pour supprimer un catalogue de données

L'`delete-data-catalog`exemple suivant supprime le catalogue de `UnusedDataCatalog` données.

```
aws athena delete-data-catalog \
    --name UnusedDataCatalog
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un catalogue : delete-data-catalog dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous DeleteDataCatalogà la section Référence des AWS CLI commandes.

`delete-named-query`

L'exemple de code suivant montre comment utiliser`delete-named-query`.

AWS CLI

Pour supprimer une requête nommée

L'`delete-named-query`exemple suivant supprime la requête nommée qui possède l'ID spécifié.

```
aws athena delete-named-query \
    --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous DeleteNamedQueryà la section Référence des AWS CLI commandes.

`delete-work-group`

L'exemple de code suivant montre comment utiliser`delete-work-group`.

AWS CLI

Pour supprimer un groupe de travail

L'`delete-work-group`exemple suivant supprime le groupe de `TeamB` travail.

```
aws athena delete-work-group \
    --work-group TeamB
```

Cette commande ne produit aucun résultat. Pour confirmer la suppression, utilisez`aws athena list-work-groups`.

Pour plus d'informations, consultez la section Gestion des groupes de travail dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous DeleteWorkGroupà la section Référence des AWS CLI commandes.

`get-data-catalog`

L'exemple de code suivant montre comment utiliser`get-data-catalog`.

AWS CLI

Pour renvoyer des informations sur un catalogue de données

L'`get-data-catalog`exemple suivant renvoie des informations sur le catalogue `dynamo_db_catalog` de données.

```
aws athena get-data-catalog \
    --name dynamo_db_catalog
```

Sortie :

```
{
    "DataCatalog": {
        "Name": "dynamo_db_catalog",
        "Description": "DynamoDB Catalog",
        "Type": "LAMBDA",
        "Parameters": {
            "catalog": "dynamo_db_catalog",
            "metadata-function": "arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda",
            "record-function": "arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda"
        }
    }
```

```
}
```

Pour plus d'informations, consultez Afficher les détails du catalogue : get-data-catalog dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetDataCatalogà la section Référence des AWS CLI commandes.

`get-database`

L'exemple de code suivant montre comment utiliser`get-database`.

AWS CLI

Pour renvoyer des informations sur une base de données dans un catalogue de données

L'`get-database`exemple suivant renvoie des informations sur la `sampledb` base de données dans le catalogue de `AwsDataCatalog` données.

```
aws athena get-database \
    --catalog-name AwsDataCatalog \
    --database-name sampledb
```

Sortie :

```
{
    "Database": {
        "Name": "sampledb",
        "Description": "Sample database",
        "Parameters": {
            "CreatedBy": "Athena",
            "EXTERNAL": "TRUE"
        }
    }
}
```

Pour plus d'informations, consultez la section Afficher les détails de la base de données : get-database dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetDatabaseà la section Référence des AWS CLI commandes.

`get-named-query`

L'exemple de code suivant montre comment utiliser`get-named-query`.

AWS CLI

Pour renvoyer une requête nommée

L'`get-named-query`exemple suivant renvoie des informations sur la requête dont l'ID est spécifié.

```
aws athena get-named-query \
    --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "NamedQuery": {
        "Name": "CloudFront Logs - SFO",
        "Description": "Shows successful GET request data for SFO",
        "Database": "default",
        "QueryString": "select date, location, browser, uri, status from
 cloudfront_logs where method = 'GET' and status = 200 and location like 'SFO%'
 limit 10",
        "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "WorkGroup": "AthenaAdmin"
    }
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetNamedQueryà la section Référence des AWS CLI commandes.

`get-query-execution`

L'exemple de code suivant montre comment utiliser`get-query-execution`.

AWS CLI

Pour renvoyer des informations sur l'exécution d'une requête

L'`get-query-execution`exemple suivant renvoie des informations sur la requête dont l'ID de requête est spécifié.

```
aws athena get-query-execution \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "QueryExecution": {
        "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Query": "select date, location, browser, uri, status from cloudfront_logs
 where method = 'GET
' and status = 200 and location like 'SFO%' limit 10",
        "StatementType": "DML",
        "ResultConfiguration": {
            "OutputLocation": "s3://amzn-s3-demo-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111.csv"
        },
        "QueryExecutionContext": {
            "Database": "mydatabase",
            "Catalog": "awsdatacatalog"
        },
        "Status": {
            "State": "SUCCEEDED",
            "SubmissionDateTime": 1593469842.665,
            "CompletionDateTime": 1593469846.486
        },
        "Statistics": {
            "EngineExecutionTimeInMillis": 3600,
            "DataScannedInBytes": 203089,
            "TotalExecutionTimeInMillis": 3821,
            "QueryQueueTimeInMillis": 267,
            "QueryPlanningTimeInMillis": 1175
        },
        "WorkGroup": "AthenaAdmin"
    }
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetQueryExecutionà la section Référence des AWS CLI commandes.

`get-query-results`

L'exemple de code suivant montre comment utiliser`get-query-results`.

AWS CLI

Pour renvoyer les résultats d'une requête

L'`get-query-results`exemple suivant renvoie les résultats de la requête dont l'ID de requête est spécifié.

```
aws athena get-query-results \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "ResultSet": {
        "Rows": [
            {
                "Data": [
                    {
                        "VarCharValue": "date"
                    },
                    {
                        "VarCharValue": "location"
                    },
                    {
                        "VarCharValue": "browser"
                    },
                    {
                        "VarCharValue": "uri"
                    },
                    {
                        "VarCharValue": "status"
                    }
                ]
            },
            {
```

```
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Safari"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Opera"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
```

```
                    },
                    {
                        "VarCharValue": "Firefox"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Lynx"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "IE"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
```

```
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Opera"
                    },
                    {
                        "VarCharValue": "/test-image-1.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Chrome"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
```

```
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Firefox"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Chrome"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
```

```
                    "VarCharValue": "SFO4"
                },
                {
                    "VarCharValue": "IE"
                },
                {
                    "VarCharValue": "/test-image-2.jpeg"
                },
                {
                    "VarCharValue": "200"
                }
            ]
        }
    ],
    "ResultSetMetadata": {
        "ColumnInfo": [
            {
                "CatalogName": "hive",
                "SchemaName": "",
                "TableName": "",
                "Name": "date",
                "Label": "date",
                "Type": "date",
                "Precision": 0,
                "Scale": 0,
                "Nullable": "UNKNOWN",
                "CaseSensitive": false
            },
            {
                "CatalogName": "hive",
                "SchemaName": "",
                "TableName": "",
                "Name": "location",
                "Label": "location",
                "Type": "varchar",
                "Precision": 2147483647,
            "Data": [

                "Scale": 0,
                "Nullable": "UNKNOWN",
                "CaseSensitive": true
            },
            {
                "CatalogName": "hive",
```

```
                    "SchemaName": "",
                    "TableName": "",
                    "Name": "browser",
                    "Label": "browser",
                    "Type": "varchar",
                    "Precision": 2147483647,
                    "Scale": 0,
                    "Nullable": "UNKNOWN",
                    "CaseSensitive": true
                },
                {
                    "CatalogName": "hive",
                    "SchemaName": "",
                    "TableName": "",
                    "Name": "uri",
                    "Label": "uri",
                    "Type": "varchar",
                    "Precision": 2147483647,
                    "Scale": 0,
                    "Nullable": "UNKNOWN",
                    "CaseSensitive": true
                },
                {
                    "CatalogName": "hive",
                    "SchemaName": "",
                    "TableName": "",
                    "Name": "status",
                    "Label": "status",
                    "Type": "integer",
                    "Precision": 10,
                    "Scale": 0,
                    "Nullable": "UNKNOWN",
                    "CaseSensitive": false
                }
            ]
        }
    },
    "UpdateCount": 0
}
```

Pour plus d'informations, consultez la section Utilisation des résultats de requête, des fichiers de sortie et de l'historique des requêtes dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetQueryResultsà la section Référence des AWS CLI commandes.

`get-table-metadata`

L'exemple de code suivant montre comment utiliser`get-table-metadata`.

AWS CLI

Pour renvoyer des informations de métadonnées relatives à une table

L'`get-table-metadata`exemple suivant renvoie des informations de métadonnées relatives à la `counties` table, notamment les noms de colonnes et leurs types de données, à partir de la `sampledb` base de données du catalogue de `AwsDataCatalog` données.

```
aws athena get-table-metadata \
    --catalog-name AwsDataCatalog \
    --database-name sampledb \
    --table-name counties
```

Sortie :

```
{
    "TableMetadata": {
        "Name": "counties",
        "CreateTime": 1593559968.0,
        "LastAccessTime": 0.0,
        "TableType": "EXTERNAL_TABLE",
        "Columns": [
            {
                "Name": "name",
                "Type": "string",
                "Comment": "from deserializer"
            },
            {
                "Name": "boundaryshape",
                "Type": "binary",
                "Comment": "from deserializer"
            },
            {
                "Name": "motto",
                "Type": "string",
```

```
                "Comment": "from deserializer"
            },
            {
                "Name": "population",
                "Type": "int",
                "Comment": "from deserializer"
            }
        ],
        "PartitionKeys": [],
        "Parameters": {
            "EXTERNAL": "TRUE",
            "inputformat": "com.esri.json.hadoop.EnclosedJsonInputFormat",
            "location": "s3://amzn-s3-demo-bucket/json",
            "outputformat":
 "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
            "serde.param.serialization.format": "1",
            "serde.serialization.lib": "com.esri.hadoop.hive.serde.JsonSerde",
            "transient_lastDdlTime": "1593559968"
        }
    }
}
```

Pour plus d'informations, consultez Afficher les détails du tableau : get-table-metadata dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetTableMetadataà la section Référence des AWS CLI commandes.

`get-work-group`

L'exemple de code suivant montre comment utiliser`get-work-group`.

AWS CLI

Pour renvoyer des informations sur un groupe de travail

L'`get-work-group`exemple suivant renvoie des informations sur le `AthenaAdmin` groupe de travail.

```
aws athena get-work-group \
    --work-group AthenaAdmin
```

Sortie :

```
{
    "WorkGroup": {
        "Name": "AthenaAdmin",
        "State": "ENABLED",
        "Configuration": {
            "ResultConfiguration": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/"
            },
            "EnforceWorkGroupConfiguration": false,
            "PublishCloudWatchMetricsEnabled": true,
            "RequesterPaysEnabled": false
        },
        "Description": "Workgroup for Athena administrators",
        "CreationTime": 1573677174.105
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de travail dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous GetWorkGroupà la section Référence des AWS CLI commandes.

`list-data-catalogs`

L'exemple de code suivant montre comment utiliser`list-data-catalogs`.

AWS CLI

Pour répertorier les catalogues de données enregistrés auprès d'Athena

L'`list-data-catalogs`exemple suivant répertorie les catalogues de données enregistrés auprès d'Athena.

```
aws athena list-data-catalogs
```

Sortie :

```
{
    "DataCatalogsSummary": [
        {
            "CatalogName": "AwsDataCatalog",
```

```
            "Type": "GLUE"
        },
        {
            "CatalogName": "cw_logs_catalog",
            "Type": "LAMBDA"
        },
        {
            "CatalogName": "cw_metrics_catalog",
            "Type": "LAMBDA"
        }
    ]
}
```

Pour plus d'informations, consultez la section Liste des catalogues enregistrés : list-data-catalogs dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListDataCatalogsà la section Référence des AWS CLI commandes.

`list-databases`

L'exemple de code suivant montre comment utiliser`list-databases`.

AWS CLI

Pour répertorier les bases de données dans un catalogue de données

L'`list-databases`exemple suivant répertorie les bases de données du catalogue de `AwsDataCatalog` données.

```
aws athena list-databases \
    --catalog-name AwsDataCatalog
```

Sortie :

```
{
    "DatabaseList": [
        {
            "Name": "default"
        },
        {
            "Name": "mydatabase"
        },
```

```
        {
            "Name": "newdb"
        },
        {
            "Name": "sampledb",
            "Description": "Sample database",
            "Parameters": {
                "CreatedBy": "Athena",
                "EXTERNAL": "TRUE"
            }
        },
        {
            "Name": "webdata"
        }
    ]
}
```

Pour plus d'informations, consultez la section Listing Databases in a Catalog : list-databases dans le guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListDatabasesà la section Référence des AWS CLI commandes.

`list-named-queries`

L'exemple de code suivant montre comment utiliser`list-named-queries`.

AWS CLI

Pour répertorier les requêtes nommées pour un groupe de travail

L'`list-named-queries`exemple suivant répertorie les requêtes nommées pour le `AthenaAdmin` groupe de travail.

```
aws athena list-named-queries \
    --work-group AthenaAdmin
```

Sortie :

```
{
    "NamedQueryIds": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
    ]
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListNamedQueriesà la section Référence des AWS CLI commandes.

`list-query-executions`

L'exemple de code suivant montre comment utiliser`list-query-executions`.

AWS CLI

Pour répertorier la requête IDs des requêtes dans un groupe de travail spécifié

L'`list-query-executions`exemple suivant répertorie un maximum de dix requêtes IDs dans le `AthenaAdmin` groupe de travail.

```
aws athena list-query-executions \
    --work-group AthenaAdmin \
    --max-items 10
```

Sortie :

```
{
    "QueryExecutionIds": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11110",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11114",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11115",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11116",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11117",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11118",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11119"
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

Pour plus d'informations, consultez la section Utilisation des résultats de requête, des fichiers de sortie et de l'historique des requêtes dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListQueryExecutionsà la section Référence des AWS CLI commandes.

`list-table-metadata`

L'exemple de code suivant montre comment utiliser`list-table-metadata`.

AWS CLI

Pour répertorier les métadonnées des tables dans la base de données spécifiée d'un catalogue de données

L'`list-table-metadata`exemple suivant renvoie des informations de métadonnées pour un maximum de deux tables de la `geography` base de données du catalogue de `AwsDataCatalog` données.

```
aws athena list-table-metadata \
    --catalog-name AwsDataCatalog \
    --database-name geography \
    --max-items 2
```

Sortie :

```
{
    "TableMetadataList": [
        {
            "Name": "country_codes",
            "CreateTime": 1586553454.0,
            "TableType": "EXTERNAL_TABLE",
            "Columns": [
                {
                    "Name": "country",
                    "Type": "string",
                    "Comment": "geo id"
                },
                {
                    "Name": "alpha-2 code",
                    "Type": "string",
                    "Comment": "geo id2"
```

```
                },
                {
                    "Name": "alpha-3 code",
                    "Type": "string",
                    "Comment": "state name"
                },
                {
                    "Name": "numeric code",
                    "Type": "bigint",
                    "Comment": ""
                },
                {
                    "Name": "latitude",
                    "Type": "bigint",
                    "Comment": "location (latitude)"
                },
                {
                    "Name": "longitude",
                    "Type": "bigint",
                    "Comment": "location (longitude)"
                }
            ],
            "Parameters": {
                "areColumnsQuoted": "false",
                "classification": "csv",
                "columnsOrdered": "true",
                "delimiter": ",",
                "has_encrypted_data": "false",
                "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
                "location": "s3://amzn-s3-demo-bucket/csv/countrycode",
                "outputformat":
"org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
                "serde.param.field.delim": ",",
                "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
                "skip.header.line.count": "1",
                "typeOfData": "file"
            }
        },
        {
            "Name": "county_populations",
            "CreateTime": 1586553446.0,
            "TableType": "EXTERNAL_TABLE",
            "Columns": [
```

```
                {
                    "Name": "id",
                    "Type": "string",
                    "Comment": "geo id"
                },
                {
                    "Name": "country",

                    "Name": "id2",
                    "Type": "string",
                    "Comment": "geo id2"
                },
                {
                    "Name": "county",
                    "Type": "string",
                    "Comment": "county name"
                },
                {
                    "Name": "state",
                    "Type": "string",
                    "Comment": "state name"
                },
                {
                    "Name": "population estimate 2018",
                    "Type": "string",
                    "Comment": ""
                }
            ],
            "Parameters": {
                "areColumnsQuoted": "false",
                "classification": "csv",
                "columnsOrdered": "true",
                "delimiter": ",",
                "has_encrypted_data": "false",
                "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
                "location": "s3://amzn-s3-demo-bucket/csv/CountyPopulation",
                "outputformat":
"org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
                "serde.param.field.delim": ",",
                "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
                "skip.header.line.count": "1",
                "typeOfData": "file"
            }
```

```
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

Pour plus d'informations, consultez Afficher les métadonnées pour toutes les tables d'une base de données : list-table-metadata dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListTableMetadataà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Exemple 1 : pour répertorier les balises d'un groupe de travail

L'`list-tags-for-resource`exemple suivant répertorie les balises du `Data_Analyst_Group` groupe de travail.

```
aws athena list-tags-for-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/
Data_Analyst_Group
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Division",
            "Value": "West"
        },
        {
            "Key": "Team",
            "Value": "Big Data"
        },
        {
            "Key": "Location",
            "Value": "Seattle"
```

```
        }
    ]
}
```

Exemple 2 : pour répertorier les balises d'un catalogue de données

L'`list-tags-for-resource`exemple suivant répertorie les balises du catalogue `dynamo_db_catalog` de données.

```
aws athena list-tags-for-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Division",
            "Value": "Mountain"
        },
        {
            "Key": "Organization",
            "Value": "Retail"
        },
        {
            "Key": "Product_Line",
            "Value": "Shoes"
        },
        {
            "Key": "Location",
            "Value": "Denver"
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les balises d'une ressource : list-tags-for-resource dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-work-groups`

L'exemple de code suivant montre comment utiliser`list-work-groups`.

AWS CLI

Pour répertorier les groupes de travail

L'`list-work-groups`exemple suivant répertorie les groupes de travail du compte courant.

```
aws athena list-work-groups
```

Sortie :

```
{
    "WorkGroups": [
        {
            "Name": "Data_Analyst_Group",
            "State": "ENABLED",
            "Description": "",
            "CreationTime": 1578006683.016
        },
        {
            "Name": "AthenaAdmin",
            "State": "ENABLED",
            "Description": "",
            "CreationTime": 1573677174.105
        },
        {
            "Name": "primary",
            "State": "ENABLED",
            "Description": "",
            "CreationTime": 1567465222.723
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des groupes de travail dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous ListWorkGroupsà la section Référence des AWS CLI commandes.

`start-query-execution`

L'exemple de code suivant montre comment utiliser`start-query-execution`.

AWS CLI

Exemple 1 : pour exécuter une requête dans un groupe de travail sur la table spécifiée dans la base de données et le catalogue de données spécifiés

L'`start-query-execution`exemple suivant utilise le `AthenaAdmin` groupe de travail pour exécuter une requête sur la `cloudfront_logs` table du catalogue `cflogsdatabase` de `AwsDataCatalog` données.

```
aws athena start-query-execution \
    --query-string "select date, location, browser, uri, status from cloudfront_logs
 where method = 'GET' and status = 200 and location like 'SFO%' limit 10" \
    --work-group "AthenaAdmin" \
    --query-execution-context Database=cflogsdatabase,Catalog=AwsDataCatalog
```

Sortie :

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

Exemple 2 : pour exécuter une requête utilisant un groupe de travail spécifique afin de créer une base de données dans le catalogue de données spécifié

L'`start-query-execution`exemple suivant utilise le `AthenaAdmin` groupe de travail pour créer la base de données `newdb` dans le catalogue `AwsDataCatalog` de données par défaut.

```
aws athena start-query-execution \
    --query-string "create database if not exists newdb" \
    --work-group "AthenaAdmin"
```

Sortie :

```
{
```

```
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

Exemple 3 : pour exécuter une requête qui crée une vue sur une table dans la base de données et le catalogue de données spécifiés

L'`start-query-execution`exemple suivant utilise une `SELECT` instruction sur le `cloudfront_logs` tableau du `cflogsdatabase` pour créer la vue`cf10`.

```
aws athena start-query-execution \
    --query-string  "CREATE OR REPLACE VIEW cf10 AS SELECT * FROM cloudfront_logs
 limit 10" \
    --query-execution-context Database=cflogsdatabase
```

Sortie :

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"
}
```

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous StartQueryExecutionà la section Référence des AWS CLI commandes.

`stop-query-execution`

L'exemple de code suivant montre comment utiliser`stop-query-execution`.

AWS CLI

Pour arrêter une requête en cours

L'`stop-query-execution`exemple suivant arrête la requête qui possède l'ID de requête spécifié.

```
aws athena stop-query-execution \
```

```
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, consultez Exécution de requêtes SQL à l'aide d'Amazon Athena dans le Guide de l'utilisateur Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous StopQueryExecutionà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Ajout d'une balise à une ressource

L'`tag-resource`exemple suivant ajoute trois balises au catalogue `dynamo_db_catalog` de données.

```
aws athena tag-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog \
    --
tags Key=Organization,Value=Retail Key=Division,Value=Mountain Key=Product_Line,Value=Shoes
```

Cette commande ne produit aucun résultat. Pour voir le résultat, utilisez`aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog`.

Pour plus d'informations, consultez la section Ajouter des balises à une ressource : tag-resource dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime les `Focus` clés `Specialization` et et leurs valeurs associées de la ressource du catalogue de `dynamo_db_catalog` données.

```
aws athena untag-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog \
    --tag-keys Specialization Focus
```

Cette commande ne produit aucun résultat. Pour voir les résultats, utilisez la `list-tags-for-resource` commande.

Pour plus d'informations, consultez Supprimer les balises d'une ressource : untag-resource dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-data-catalog`

L'exemple de code suivant montre comment utiliser`update-data-catalog`.

AWS CLI

Pour mettre à jour un catalogue de données

L'`update-data-catalog`exemple suivant met à jour la fonction Lambda et la description du catalogue de `cw_logs_catalog` données.

```
aws athena update-data-catalog \
    --name cw_logs_catalog \
    --type LAMBDA \
    --description "New CloudWatch Logs Catalog" \
    --function=arn:aws:lambda:us-west-2:111122223333:function:new_cw_logs_lambda
```

Cette commande ne produit aucun résultat. Pour voir le résultat, utilisez`aws athena get-data-catalog --name cw_logs_catalog`.

Pour plus d'informations, consultez Mettre à jour un catalogue : update-data-catalog dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous UpdateDataCatalogà la section Référence des AWS CLI commandes.

`update-work-group`

L'exemple de code suivant montre comment utiliser`update-work-group`.

AWS CLI

Pour mettre à jour un groupe de travail

L'`update-work-group`exemple suivant désactive le groupe de `Data_Analyst_Group` travail. Les utilisateurs ne peuvent pas exécuter ou créer de requêtes dans le groupe de travail désactivé, mais peuvent toujours consulter les métriques, les contrôles des limites d'utilisation des données, les paramètres du groupe de travail, l'historique des requêtes et les requêtes enregistrées.

```
aws athena update-work-group \
    --work-group Data_Analyst_Group \
    --state DISABLED
```

Cette commande ne produit aucun résultat. Pour vérifier le changement d'état, utilisez `aws athena get-work-group --work-group Data_Analyst_Group` et vérifiez la `State` propriété dans la sortie.

Pour plus d'informations, consultez la section Gestion des groupes de travail dans le guide de l'utilisateur d'Amazon Athena.

- Pour plus de détails sur l'API, reportez-vous UpdateWorkGroupà la section Référence des AWS CLI commandes.

Exemples d'Auto Scaling utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Auto Scaling.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`attach-instances`

L'exemple de code suivant montre comment utiliser`attach-instances`.

AWS CLI

Pour attacher une instance à un groupe Auto Scaling

Cet exemple attache l'instance spécifiée au groupe Auto Scaling spécifié.

```
aws autoscaling attach-instances \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AttachInstancesà la section Référence des AWS CLI commandes.

`attach-load-balancer-target-groups`

L'exemple de code suivant montre comment utiliser`attach-load-balancer-target-groups`.

AWS CLI

Pour associer un groupe cible à un groupe Auto Scaling

Cet exemple attache le groupe cible spécifié au groupe Auto Scaling spécifié.

```
aws autoscaling attach-load-balancer-target-groups \
    --auto-scaling-group-name my-asg \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Elastic Load Balancing et Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous AttachLoadBalancerTargetGroupsà la section Référence des AWS CLI commandes.

`attach-load-balancers`

L'exemple de code suivant montre comment utiliser`attach-load-balancers`.

AWS CLI

Pour associer un Classic Load Balancer à un groupe Auto Scaling

Cet exemple attache le Classic Load Balancer spécifié au groupe Auto Scaling spécifié.

```
aws autoscaling attach-load-balancers \
    --load-balancer-names my-load-balancer \
    --auto-scaling-group-name my-asg
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Elastic Load Balancing et Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous AttachLoadBalancersà la section Référence des AWS CLI commandes.

`cancel-instance-refresh`

L'exemple de code suivant montre comment utiliser`cancel-instance-refresh`.

AWS CLI

Pour annuler l'actualisation d'une instance

L'`cancel-instance-refresh`exemple suivant annule une actualisation d'instance en cours pour le groupe Auto Scaling spécifié.

```
aws autoscaling cancel-instance-refresh \
```

```
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Pour plus d'informations, consultez Annuler l'actualisation d'une instance dans le manuel Amazon EC2 Auto Scaling User Guide.

- Pour plus de détails sur l'API, reportez-vous CancelInstanceRefreshà la section Référence des AWS CLI commandes.

`complete-lifecycle-action`

L'exemple de code suivant montre comment utiliser`complete-lifecycle-action`.

AWS CLI

Pour terminer l'action du cycle de vie

Cet exemple indique à Amazon EC2 Auto Scaling que l'action du cycle de vie spécifiée est terminée afin qu'il puisse terminer le lancement ou la fin de l'instance.

```
aws autoscaling complete-lifecycle-action \
    --lifecycle-hook-name my-launch-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-action-result CONTINUE \
    --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous CompleteLifecycleActionà la section Référence des AWS CLI commandes.

`create-auto-scaling-group`

L'exemple de code suivant montre comment utiliser`create-auto-scaling-group`.

AWS CLI

Exemple 1 : pour créer un groupe Auto Scaling

L'`create-auto-scaling-group`exemple suivant crée un groupe Auto Scaling dans des sous-réseaux de plusieurs zones de disponibilité au sein d'une région. Les instances sont lancées avec la version par défaut du modèle de lancement spécifié. Notez que les valeurs par défaut sont utilisées pour la plupart des autres paramètres, tels que les politiques de résiliation et la configuration du bilan de santé.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12 \
    --min-size 1 \
    --max-size 5 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour associer un Application Load Balancer, un Network Load Balancer ou un Gateway Load Balancer

Cet exemple indique l'ARN d'un groupe cible pour un équilibreur de charge qui prend en charge le trafic attendu. Le type de bilan de santé indique `ELB` que lorsqu'Elastic Load Balancing signale qu'une instance est défectueuse, le groupe Auto Scaling la remplace. La commande définit également un délai de grâce en `600` secondes pour le contrôle de santé. Le délai de grâce permet d'éviter la résiliation prématurée des instances nouvellement lancées.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12 \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/943f017f100becff \
    --health-check-type ELB \
    --health-check-grace-period 600 \
    --min-size 1 \
    --max-size 5 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Elastic Load Balancing et Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 3 : pour spécifier un groupe de placement et utiliser la dernière version du modèle de lancement

Cet exemple lance des instances dans un groupe de placement au sein d'une seule zone de disponibilité. Cela peut être utile pour les groupes à faible latence soumis à des charges de travail HPC. Cet exemple indique également la taille minimale, la taille maximale et la capacité souhaitée du groupe.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \
    --min-size 1 \
    --max-size 5 \
    --desired-capacity 3 \
    --placement-group my-placement-group \
    --vpc-zone-identifier "subnet-6194ea3b"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes de placement dans le Guide de EC2 l'utilisateur Amazon pour les instances Linux.

Exemple 4 : Pour spécifier une instance unique, le groupe Auto Scaling et utiliser une version spécifique du modèle de lancement

Cet exemple crée un groupe Auto Scaling dont les capacités minimale et maximale sont définies 1 de manière à garantir l'exécution d'une instance. La commande spécifie également la v1 d'un modèle de lancement dans lequel l'ID d'un ENI existant est spécifié. Lorsque vous utilisez un modèle de lancement qui spécifie une ENI existante pour eth0, vous devez spécifier une zone de disponibilité pour le groupe Auto Scaling qui correspond à l'interface réseau, sans également spécifier d'ID de sous-réseau dans la demande.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-single-instance \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1' \
    --min-size 1 \
```

```
    --max-size 1 \
    --availability-zones us-west-2a
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 5 : pour spécifier une politique de résiliation différente

Cet exemple crée un groupe Auto Scaling à l'aide d'une configuration de lancement et définit la politique de résiliation pour mettre fin aux instances les plus anciennes en premier. La commande applique également une balise au groupe et à ses instances, avec une clé `Role` et une valeur de`WebServer`.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-configuration-name my-lc \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "OldestInstance" \
    --tags "ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=Role,Value=WebServer,PropagateAtLaunch=true" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Working with Amazon EC2 Auto Scaling résiliation policies dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

Exemple 6 : pour spécifier un hook du cycle de vie de lancement

Cet exemple crée un groupe Auto Scaling avec un hook de cycle de vie qui prend en charge une action personnalisée lors du lancement de l'instance.

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

Contenu du `config.json` fichier :

```
{
```

```
    "AutoScalingGroupName": "my-asg",
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-1234567890abcde12"
    },
    "LifecycleHookSpecificationList": [{
        "LifecycleHookName": "my-launch-hook",
        "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
        "NotificationTargetARN": "arn:aws:sqs:us-west-2:123456789012:my-sqs-queue",
        "RoleARN": "arn:aws:iam::123456789012:role/my-notification-role",
        "NotificationMetadata": "SQS message metadata",
        "HeartbeatTimeout": 4800,
        "DefaultResult": "ABANDON"
    }],
    "MinSize": 1,
    "MaxSize": 5,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": [{
        "ResourceType": "auto-scaling-group",
        "ResourceId": "my-asg",
        "PropagateAtLaunch": true,
        "Value": "test",
        "Key": "environment"
    }]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 7 : Pour spécifier un hook du cycle de vie de terminaison

Cet exemple crée un groupe Auto Scaling avec un hook de cycle de vie qui prend en charge une action personnalisée lors de la fermeture de l'instance.

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

Contenu de `config.json` :

```
{
    "AutoScalingGroupName": "my-asg",
```

```
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-1234567890abcde12"
    },
    "LifecycleHookSpecificationList": [{
        "LifecycleHookName": "my-termination-hook",
        "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
        "HeartbeatTimeout": 120,
        "DefaultResult": "CONTINUE"
    }],
    "MinSize": 1,
    "MaxSize": 5,
    "TargetGroupARNs": [
        "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067"
    ],
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 8 : pour définir une politique de résiliation personnalisée

Cet exemple crée un groupe Auto Scaling qui spécifie une politique d'arrêt de fonction Lambda personnalisée qui indique à Amazon EC2 Auto Scaling quelles instances peuvent être interrompues en toute sécurité à grande échelle.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-single-instance \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "arn:aws:lambda:us-
west-2:123456789012:function:HelloFunction:prod" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une politique de résiliation personnalisée avec Lambda dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous CreateAutoScalingGroupà la section Référence des AWS CLI commandes.

`create-launch-configuration`

L'exemple de code suivant montre comment utiliser`create-launch-configuration`.

AWS CLI

Exemple 1 : pour créer une configuration de lancement

Cet exemple crée une configuration de lancement simple.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une configuration de lancement dans le manuel Amazon EC2 Auto Scaling User Guide.

Exemple 2 : pour créer une configuration de lancement avec un groupe de sécurité, une paire de clés et un script de démarrage

Cet exemple crée une configuration de lancement avec un groupe de sécurité, une paire de clés et un script de démarrage contenus dans les données utilisateur.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --security-groups sg-eb2af88example \
    --key-name my-key-pair \
    --user-data file://myuserdata.txt
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une configuration de lancement dans le manuel Amazon EC2 Auto Scaling User Guide.

Exemple 3 : pour créer une configuration de lancement avec un rôle IAM

Cet exemple crée une configuration de lancement avec le nom de profil d'instance d'un rôle IAM.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --iam-instance-profile my-autoscaling-role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez le rôle IAM pour les applications qui s'exécutent sur des EC2 instances Amazon dans le manuel Amazon EC2 Auto Scaling User Guide.

Exemple 4 : pour créer une configuration de lancement avec la surveillance détaillée activée

Cet exemple crée une configuration de lancement avec la surveillance EC2 détaillée activée, qui envoie EC2 des métriques par CloudWatch intervalles d'une minute.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --instance-monitoring Enabled=true
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration de la surveillance pour les instances Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 5 : pour créer une configuration de lancement qui lance des instances Spot

Cet exemple crée une configuration de lancement qui utilise les instances Spot comme seule option d'achat.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --spot-price "0.50"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Request Spot Instances dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 6 : pour créer une configuration de lancement à l'aide d'une EC2 instance

Cet exemple crée une configuration de lancement basée sur les attributs d'une instance existante. Il remplace la location de placement et indique si une adresse IP publique est définie en incluant les options `--placement-tenancy` et`--no-associate-public-ip-address`.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc-from-instance \
    --instance-id i-0123a456700123456 \
    --instance-type m5.large \
    --no-associate-public-ip-address \
    --placement-tenancy dedicated
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une configuration de lancement à l'aide d'une EC2 instance dans le manuel Amazon EC2 Auto Scaling User Guide.

Exemple 7 : pour créer une configuration de lancement avec un mappage de périphériques en mode bloc pour un volume Amazon EBS

Cet exemple crée une configuration de lancement avec un mappage de périphériques par blocs pour un gp3 volume Amazon EBS avec le nom de l'appareil `/dev/sdh` et une taille de volume de 20.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --block-device-mappings '[{"DeviceName":"/dev/sdh","Ebs":
{"VolumeSize":20,"VolumeType":"gp3"}}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez EBS dans le manuel Amazon EC2 Auto Scaling API Reference.

Pour plus d'informations sur la syntaxe permettant de citer des valeurs de paramètres au format JSON, consultez la section Utilisation de guillemets avec des chaînes dans la AWS CLI du Guide de l'utilisateur de l'interface de ligne de AWS commande.

Exemple 8 : pour créer une configuration de lancement avec un mappage de périphériques en mode bloc pour un volume de stockage d'instance

Cet exemple crée une configuration de lancement avec `ephemeral1` comme instance un volume de stockage avec le nom du périphérique`/dev/sdc`.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --block-device-mappings '[{"DeviceName":"/dev/sdc","VirtualName":"ephemeral1"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez BlockDeviceMappingle manuel Amazon EC2 Auto Scaling API Reference.

Pour plus d'informations sur la syntaxe permettant de citer des valeurs de paramètres au format JSON, consultez la section Utilisation de guillemets avec des chaînes dans la AWS CLI du Guide de l'utilisateur de l'interface de ligne de AWS commande.

Exemple 9 : pour créer une configuration de lancement et empêcher un périphérique bloqué de se connecter au moment du lancement

Cet exemple crée une configuration de lancement qui supprime un périphérique en mode bloc spécifié par le mappage des périphériques en mode bloc de l'AMI (par exemple,`/dev/sdf`).

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --block-device-mappings '[{"DeviceName":"/dev/sdf","NoDevice":""}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez BlockDeviceMappingle manuel Amazon EC2 Auto Scaling API Reference.

Pour plus d'informations sur la syntaxe permettant de citer des valeurs de paramètres au format JSON, consultez la section Utilisation de guillemets avec des chaînes dans la AWS CLI du Guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous CreateLaunchConfigurationà la section Référence des AWS CLI commandes.

`create-or-update-tags`

L'exemple de code suivant montre comment utiliser`create-or-update-tags`.

AWS CLI

Pour créer ou mettre à jour des balises pour un groupe Auto Scaling

Cet exemple ajoute deux balises au groupe Auto Scaling spécifié.

```
aws autoscaling create-or-update-tags \
    --tags ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=Role,Value=WebServer,PropagateAtLaunch=true ResourceId=my-
asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research,PropagateAtLaunch=true
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Tagging Auto Scaling groups and instances dans le Amazon EC2 Auto Scaling User Guide.

- Pour plus de détails sur l'API, reportez-vous CreateOrUpdateTagsà la section Référence des AWS CLI commandes.

`delete-auto-scaling-group`

L'exemple de code suivant montre comment utiliser`delete-auto-scaling-group`.

AWS CLI

Exemple 1 : pour supprimer le groupe Auto Scaling spécifié

Cet exemple supprime le groupe Auto Scaling spécifié.

```
aws autoscaling delete-auto-scaling-group \
```

```
    --auto-scaling-group-name my-asg
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer votre infrastructure Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : Pour forcer la suppression du groupe Auto Scaling spécifié

Pour supprimer le groupe Auto Scaling sans attendre que les instances du groupe se terminent, utilisez l'`--force-delete`option.

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --force-delete
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer votre infrastructure Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DeleteAutoScalingGroupà la section Référence des AWS CLI commandes.

`delete-launch-configuration`

L'exemple de code suivant montre comment utiliser`delete-launch-configuration`.

AWS CLI

Pour supprimer une configuration de lancement

Cet exemple supprime la configuration de lancement spécifiée.

```
aws autoscaling delete-launch-configuration \
    --launch-configuration-name my-launch-config
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer votre infrastructure Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DeleteLaunchConfigurationà la section Référence des AWS CLI commandes.

`delete-lifecycle-hook`

L'exemple de code suivant montre comment utiliser`delete-lifecycle-hook`.

AWS CLI

Pour supprimer un hook du cycle de vie

Cet exemple supprime le hook de cycle de vie spécifié.

```
aws autoscaling delete-lifecycle-hook \
    --lifecycle-hook-name my-lifecycle-hook \
    --auto-scaling-group-name my-asg
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteLifecycleHookà la section Référence des AWS CLI commandes.

`delete-notification-configuration`

L'exemple de code suivant montre comment utiliser`delete-notification-configuration`.

AWS CLI

Pour supprimer une notification Auto Scaling

Cet exemple supprime la notification spécifiée du groupe Auto Scaling spécifié.

```
aws autoscaling delete-notification-configuration \
    --auto-scaling-group-name my-asg \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer la configuration des notifications dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DeleteNotificationConfigurationà la section Référence des AWS CLI commandes.

`delete-policy`

L'exemple de code suivant montre comment utiliser`delete-policy`.

AWS CLI

Pour supprimer une politique de dimensionnement

Cet exemple supprime la politique de dimensionnement spécifiée.

```
aws autoscaling delete-policy \
    --auto-scaling-group-name my-asg \
    --policy-name alb1000-target-tracking-scaling-policy
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`delete-scheduled-action`

L'exemple de code suivant montre comment utiliser`delete-scheduled-action`.

AWS CLI

Pour supprimer une action planifiée d'un groupe Auto Scaling

Cet exemple supprime l'action planifiée spécifiée du groupe Auto Scaling spécifié.

```
aws autoscaling delete-scheduled-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-scheduled-action
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteScheduledActionà la section Référence des AWS CLI commandes.

`delete-tags`

L'exemple de code suivant montre comment utiliser`delete-tags`.

AWS CLI

Pour supprimer une balise d'un groupe Auto Scaling

Cet exemple supprime la balise spécifiée du groupe Auto Scaling spécifié.

```
aws autoscaling delete-tags \
    --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Tagging Auto Scaling groups and instances dans le Amazon EC2 Auto Scaling User Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteTagsà la section Référence des AWS CLI commandes.

`delete-warm-pool`

L'exemple de code suivant montre comment utiliser`delete-warm-pool`.

AWS CLI

Exemple 1 : pour supprimer un pool de chaleur

L'exemple suivant supprime le pool de chaleur pour le groupe Auto Scaling spécifié.

```
aws autoscaling delete-warm-pool \
    --auto-scaling-group-name my-asg
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Warm pools for Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : Pour forcer la suppression d'un pool de chaleur

Pour supprimer le pool de chaleur sans attendre que ses instances se terminent, utilisez l'`--force-delete`option.

```
aws autoscaling delete-warm-pool \
```

```
    --auto-scaling-group-name my-asg \
    --force-delete
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Warm pools for Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DeleteWarmPoolà la section Référence des AWS CLI commandes.

`describe-account-limits`

L'exemple de code suivant montre comment utiliser`describe-account-limits`.

AWS CLI

Pour décrire les limites de votre compte Amazon EC2 Auto Scaling

Cet exemple décrit les limites d'Amazon EC2 Auto Scaling pour votre AWS compte.

```
aws autoscaling describe-account-limits
```

Sortie :

```
{
    "NumberOfLaunchConfigurations": 5,
    "MaxNumberOfLaunchConfigurations": 100,
    "NumberOfAutoScalingGroups": 3,
    "MaxNumberOfAutoScalingGroups": 20
}
```

Pour plus d'informations, consultez les quotas du service Amazon EC2 Auto Scaling dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeAccountLimitsà la section Référence des AWS CLI commandes.

`describe-adjustment-types`

L'exemple de code suivant montre comment utiliser`describe-adjustment-types`.

AWS CLI

Pour décrire les types de réglage de mise à l'échelle disponibles

Cet exemple décrit les types de réglage disponibles.

```
aws autoscaling describe-adjustment-types
```

Sortie :

```
{
    "AdjustmentTypes": [
        {
            "AdjustmentType": "ChangeInCapacity"
        },
        {
            "AdjustmentType": "ExactCapacity"
        },
        {
            "AdjustmentType": "PercentChangeInCapacity"
        }
    ]
}
```

Pour plus d'informations, consultez la section Types d'ajustement de dimensionnement dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeAdjustmentTypesà la section Référence des AWS CLI commandes.

`describe-auto-scaling-groups`

L'exemple de code suivant montre comment utiliser`describe-auto-scaling-groups`.

AWS CLI

Exemple 1 : Pour décrire le groupe Auto Scaling spécifié

Cet exemple décrit le groupe Auto Scaling spécifié.

```
aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-names my-asg
```

Sortie :


```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-1234567890abcde12"
            },
            "MinSize": 0,
            "MaxSize": 1,
            "DesiredCapacity": 1,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 0,
            "Instances": [
                {
                    "InstanceId": "i-06905f55584de02da",
                    "InstanceType": "t2.micro",
                    "AvailabilityZone": "us-west-2a",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
                    "ProtectedFromScaleIn": false,
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-1234567890abcde12"
                    }
                }
            ],
            "CreatedTime": "2023-10-28T02:39:22.152Z",
            "SuspendedProcesses": [],
```

```
            "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
            "EnabledMetrics": [],
            "Tags": [],
            "TerminationPolicies": [
                "Default"
            ],
            "NewInstancesProtectedFromScaleIn": false,
            "ServiceLinkedRoleARN":"arn",
            "TrafficSources": []
        }
    ]
}
```

Exemple 2 : Pour décrire les 100 premiers groupes Auto Scaling spécifiés

Cet exemple décrit les groupes Auto Scaling spécifiés. Il vous permet de spécifier jusqu'à 100 noms de groupes.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 100 \
    --auto-scaling-group-names "group1" "group2" "group3" "group4"
```

Voir l'exemple 1 pour un exemple de sortie.

Exemple 3 : Pour décrire un groupe Auto Scaling dans la région spécifiée

Cet exemple décrit les groupes Auto Scaling dans la région spécifiée, jusqu'à un maximum de 75 groupes.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 75 \
    --region us-east-1
```

Voir l'exemple 1 pour un exemple de sortie.

Exemple 4 : Pour décrire le nombre spécifié de groupes Auto Scaling

Pour renvoyer un nombre spécifique de groupes Auto Scaling, utilisez l'`--max-items`option.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 1
```

Voir l'exemple 1 pour un exemple de sortie.

Si la sortie inclut un `NextToken` champ, il existe d'autres groupes. Pour obtenir les groupes supplémentaires, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel suivant comme suit.

```
aws autoscaling describe-auto-scaling-groups \
    --starting-token Z3M3LMPEXAMPLE
```

Voir l'exemple 1 pour un exemple de sortie.

Exemple 5 : Pour décrire les groupes Auto Scaling qui utilisent des configurations de lancement

Cet exemple utilise l'`--query`option pour décrire les groupes Auto Scaling qui utilisent des configurations de lancement.

```
aws autoscaling describe-auto-scaling-groups \
    --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Sortie :

```
[
    {
        "AutoScalingGroupName": "my-asg",
        "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
        "LaunchConfigurationName": "my-lc",
        "MinSize": 0,
        "MaxSize": 1,
        "DesiredCapacity": 1,
        "DefaultCooldown": 300,
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c"
        ],
        "LoadBalancerNames": [],
        "TargetGroupARNs": [],
        "HealthCheckType": "EC2",
        "HealthCheckGracePeriod": 0,
        "Instances": [
```

```
            {
                "InstanceId": "i-088c57934a6449037",
                "InstanceType": "t2.micro",
                "AvailabilityZone": "us-west-2c",
                "HealthStatus": "Healthy",
                "LifecycleState": "InService",
                "LaunchConfigurationName": "my-lc",
                "ProtectedFromScaleIn": false
            }
        ],
        "CreatedTime": "2023-10-28T02:39:22.152Z",
        "SuspendedProcesses": [],
        "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
        "EnabledMetrics": [],
        "Tags": [],
        "TerminationPolicies": [
            "Default"
        ],
        "NewInstancesProtectedFromScaleIn": false,
        "ServiceLinkedRoleARN":"arn",
        "TrafficSources": []
    }
]
```

Pour plus d'informations, consultez la section Filtrer la sortie de la AWS CLI dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DescribeAutoScalingGroupsà la section Référence des AWS CLI commandes.

`describe-auto-scaling-instances`

L'exemple de code suivant montre comment utiliser`describe-auto-scaling-instances`.

AWS CLI

Exemple 1 : pour décrire une ou plusieurs instances

Cet exemple décrit l'instance spécifiée.

```
aws autoscaling describe-auto-scaling-instances \
    --instance-ids i-06905f55584de02da
```

Sortie :

```
{
    "AutoScalingInstances": [
        {
            "InstanceId": "i-06905f55584de02da",
            "InstanceType": "t2.micro",
            "AutoScalingGroupName": "my-asg",
            "AvailabilityZone": "us-west-2b",
            "LifecycleState": "InService",
            "HealthStatus": "HEALTHY",
            "ProtectedFromScaleIn": false,
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-1234567890abcde12",
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            }
        }
    ]
}
```

Exemple 2 : pour décrire une ou plusieurs instances

Cet exemple utilise l'`--max-items`option pour spécifier le nombre d'instances à renvoyer avec cet appel.

```
aws autoscaling describe-auto-scaling-instances \
    --max-items 1
```

Si la sortie inclut un `NextToken` champ, il existe d'autres instances. Pour obtenir les instances supplémentaires, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel suivant comme suit.

```
aws autoscaling describe-auto-scaling-instances \
    --starting-token Z3M3LMPEXAMPLE
```

Voir l’exemple 1 pour un exemple de sortie.

Exemple 3 : pour décrire les instances qui utilisent des configurations de lancement

Cet exemple utilise l'`--query`option pour décrire les instances qui utilisent des configurations de lancement.

```
aws autoscaling describe-auto-scaling-instances \
    --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

Sortie :

```
[
    {
        "InstanceId": "i-088c57934a6449037",
        "InstanceType": "t2.micro",
        "AutoScalingGroupName": "my-asg",
        "AvailabilityZone": "us-west-2c",
        "LifecycleState": "InService",
        "HealthStatus": "HEALTHY",
        "LaunchConfigurationName": "my-lc",
        "ProtectedFromScaleIn": false
    }
]
```

Pour plus d'informations, consultez la section Filtrer la sortie de la AWS CLI dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DescribeAutoScalingInstancesà la section Référence des AWS CLI commandes.

`describe-auto-scaling-notification-types`

L'exemple de code suivant montre comment utiliser`describe-auto-scaling-notification-types`.

AWS CLI

Pour décrire les types de notifications disponibles

Cet exemple décrit les types de notifications disponibles.

```
aws autoscaling describe-auto-scaling-notification-types
```

Sortie :

```
{
```

```
    "AutoScalingNotificationTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCH",
        "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
        "autoscaling:EC2_INSTANCE_TERMINATE",
        "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
        "autoscaling:TEST_NOTIFICATION"
    ]
}
```

Pour plus d'informations, consultez la section Recevoir des notifications Amazon SNS lorsque votre groupe Auto Scaling évolue dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeAutoScalingNotificationTypesà la section Référence des AWS CLI commandes.

`describe-instance-refreshes`

L'exemple de code suivant montre comment utiliser`describe-instance-refreshes`.

AWS CLI

Pour décrire les actualisations d'instances

L'`describe-instance-refreshes`exemple suivant renvoie une description de toutes les demandes d'actualisation d'instance pour le groupe Auto Scaling spécifié, y compris le message d'état et (si disponible) la raison du statut.

```
aws autoscaling describe-instance-refreshes \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "InstanceRefreshes": [
        {
            "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",
            "AutoScalingGroupName": "my-asg",
            "Status": "InProgress",
            "StatusReason": "Waiting for instances to warm up before continuing. For
 example: 0e69cc3f05f825f4f is warming up.",
            "EndTime": "2023-03-23T16:42:55Z",
```

```
            "PercentageComplete": 0,
            "InstancesToUpdate": 0,
    "Preferences": {
                "MinHealthyPercentage": 100,
                "InstanceWarmup": 300,
                "CheckpointPercentages": [
                    50
                ],
                "CheckpointDelay": 3600,
                "SkipMatching": false,
                "AutoRollback": true,
                "ScaleInProtectedInstances": "Ignore",
                "StandbyInstances": "Ignore"
            }
        },
        {
            "InstanceRefreshId": "dd7728d0-5bc4-4575-96a3-1b2c52bf8bb1",
            "AutoScalingGroupName": "my-asg",
            "Status": "Successful",
            "EndTime": "2022-06-02T16:53:37Z",
            "PercentageComplete": 100,
            "InstancesToUpdate": 0,
    "Preferences": {
                "MinHealthyPercentage": 90,
                "InstanceWarmup": 300,
                "SkipMatching": true,
                "AutoRollback": true,
                "ScaleInProtectedInstances": "Ignore",
                "StandbyInstances": "Ignore"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Vérifier l'état de l'actualisation d'une instance dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceRefreshesà la section Référence des AWS CLI commandes.

`describe-launch-configurations`

L'exemple de code suivant montre comment utiliser`describe-launch-configurations`.

AWS CLI

Exemple 1 : pour décrire la configuration de lancement spécifiée

Cet exemple décrit la configuration de lancement spécifiée.

```
aws autoscaling describe-launch-configurations \
    --launch-configuration-names my-launch-config
```

Sortie :

```
{
    "LaunchConfigurations": [
        {
            "LaunchConfigurationName": "my-launch-config",
            "LaunchConfigurationARN": "arn:aws:autoscaling:us-
west-2:123456789012:launchConfiguration:98d3b196-4cf9-4e88-8ca1-8547c24ced8b:launchConfigura
my-launch-config",
            "ImageId": "ami-0528a5175983e7f28",
            "KeyName": "my-key-pair-uswest2",
            "SecurityGroups": [
                "sg-05eaec502fcdadc2e"
            ],
            "ClassicLinkVPCSecurityGroups": [],
            "UserData": "",
            "InstanceType": "t2.micro",
            "KernelId": "",
            "RamdiskId": "",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/xvda",
                    "Ebs": {
                        "SnapshotId": "snap-06c1606ba5ca274b1",
                        "VolumeSize": 8,
                        "VolumeType": "gp2",
                        "DeleteOnTermination": true,
                        "Encrypted": false
                    }
                }
            ],
            "InstanceMonitoring": {
                "Enabled": true
            },
```

```
            "CreatedTime": "2020-10-28T02:39:22.321Z",
            "EbsOptimized": false,
            "AssociatePublicIpAddress": true,
            "MetadataOptions": {
                "HttpTokens": "required",
                "HttpPutResponseHopLimit": 1,
                "HttpEndpoint": "disabled"
            }
        }
    ]
}
```

Exemple 2 : pour décrire un nombre spécifié de configurations de lancement

Pour renvoyer un nombre spécifique de configurations de lancement, utilisez l'`--max-items`option.

```
aws autoscaling describe-launch-configurations \
    --max-items 1
```

Si la sortie inclut un `NextToken` champ, il existe d'autres configurations de lancement. Pour obtenir les configurations de lancement supplémentaires, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel suivant, comme suit.

```
aws autoscaling describe-launch-configurations \
    --starting-token Z3M3LMPEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous DescribeLaunchConfigurationsà la section Référence des AWS CLI commandes.

`describe-lifecycle-hook-types`

L'exemple de code suivant montre comment utiliser`describe-lifecycle-hook-types`.

AWS CLI

Pour décrire les types de crochets de cycle de vie disponibles

Cet exemple décrit les types de hooks de cycle de vie disponibles.

```
aws autoscaling describe-lifecycle-hook-types
```

Sortie :

```
{
    "LifecycleHookTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCHING",
        "autoscaling:EC2_INSTANCE_TERMINATING"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLifecycleHookTypesà la section Référence des AWS CLI commandes.

`describe-lifecycle-hooks`

L'exemple de code suivant montre comment utiliser`describe-lifecycle-hooks`.

AWS CLI

Pour décrire vos crochets liés au cycle de vie

Cet exemple décrit les hooks du cycle de vie pour le groupe Auto Scaling spécifié.

```
aws autoscaling describe-lifecycle-hooks \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "LifecycleHooks": [
        {
            "GlobalTimeout": 3000,
            "HeartbeatTimeout": 30,
            "AutoScalingGroupName": "my-asg",
            "LifecycleHookName": "my-launch-hook",
            "DefaultResult": "ABANDON",
            "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING"
        },
        {
            "GlobalTimeout": 6000,
            "HeartbeatTimeout": 60,
            "AutoScalingGroupName": "my-asg",
            "LifecycleHookName": "my-termination-hook",
```

```
            "DefaultResult": "CONTINUE",
            "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLifecycleHooksà la section Référence des AWS CLI commandes.

`describe-load-balancer-target-groups`

L'exemple de code suivant montre comment utiliser`describe-load-balancer-target-groups`.

AWS CLI

Pour décrire les groupes cibles d'équilibreurs de charge pour un groupe Auto Scaling

Cet exemple décrit les groupes cibles d'équilibreurs de charge attachés au groupe Auto Scaling spécifié.

```
aws autoscaling describe-load-balancer-target-groups \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "LoadBalancerTargetGroups": [
        {
            "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "State": "Added"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancerTargetGroupsà la section Référence des AWS CLI commandes.

`describe-load-balancers`

L'exemple de code suivant montre comment utiliser`describe-load-balancers`.

AWS CLI

Pour décrire les équilibreurs de charge classiques pour un groupe Auto Scaling

Cet exemple décrit les équilibreurs de charge classiques pour le groupe Auto Scaling spécifié.

```
aws autoscaling describe-load-balancers \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "LoadBalancers": [
        {
            "State": "Added",
            "LoadBalancerName": "my-load-balancer"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancersà la section Référence des AWS CLI commandes.

`describe-metric-collection-types`

L'exemple de code suivant montre comment utiliser`describe-metric-collection-types`.

AWS CLI

Pour décrire les types de collecte de mesures disponibles

Cet exemple décrit les types de collecte de métriques disponibles.

```
aws autoscaling describe-metric-collection-types
```

Sortie :

```
{
    "Metrics": [
        {
            "Metric": "GroupMinSize"
        },
```

```
        {
            "Metric": "GroupMaxSize"
        },
        {
            "Metric": "GroupDesiredCapacity"
        },
        {
            "Metric": "GroupInServiceInstances"
        },
        {
            "Metric": "GroupInServiceCapacity"
        },
        {
            "Metric": "GroupPendingInstances"
        },
        {
            "Metric": "GroupPendingCapacity"
        },
        {
            "Metric": "GroupTerminatingInstances"
        },
        {
            "Metric": "GroupTerminatingCapacity"
        },
        {
            "Metric": "GroupStandbyInstances"
        },
        {
            "Metric": "GroupStandbyCapacity"
        },
        {
            "Metric": "GroupTotalInstances"
        },
        {
            "Metric": "GroupTotalCapacity"
        }
    ],
    "Granularities": [
        {
            "Granularity": "1Minute"
        }
    ]
}
```

Pour plus d'informations, consultez les métriques de groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeMetricCollectionTypesà la section Référence des AWS CLI commandes.

`describe-notification-configurations`

L'exemple de code suivant montre comment utiliser`describe-notification-configurations`.

AWS CLI

Exemple 1 : pour décrire les configurations de notification d'un groupe spécifié

Cet exemple décrit les configurations de notification pour le groupe Auto Scaling spécifié.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "NotificationConfigurations": [
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
        },
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
        }
    ]
}
```

Pour plus d'informations, consultez la section Recevoir des notifications Amazon SNS lorsque votre groupe Auto Scaling évolue dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 1 : pour décrire un nombre spécifié de configurations de notification

Pour renvoyer un nombre spécifique de configurations de notification, utilisez le `max-items` paramètre.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-auto-scaling-group \
    --max-items 1
```

Sortie :

```
{
    "NotificationConfigurations": [
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
        },
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
        }
    ]
}
```

Si la sortie inclut un `NextToken` champ, il existe d'autres configurations de notification. Pour obtenir les configurations de notification supplémentaires, utilisez la valeur de ce champ avec le `starting-token` paramètre lors d'un appel suivant comme suit.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

Pour plus d'informations, consultez la section Recevoir des notifications Amazon SNS lorsque votre groupe Auto Scaling évolue dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeNotificationConfigurationsà la section Référence des AWS CLI commandes.

`describe-policies`

L'exemple de code suivant montre comment utiliser`describe-policies`.

AWS CLI

Exemple 1 : pour décrire les politiques de dimensionnement d'un groupe spécifié

Cet exemple décrit les politiques de dimensionnement pour le groupe Auto Scaling spécifié.

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "ScalingPolicies": [
        {
            "AutoScalingGroupName": "my-asg",
            "PolicyName": "alb1000-target-tracking-scaling-policy",
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:3065d9c8-9969-4bec-
bb6a-3fbe5550fde6:autoScalingGroupName/my-asg:policyName/alb1000-target-tracking-
scaling-policy",
            "PolicyType": "TargetTrackingScaling",
            "StepAdjustments": [],
            "Alarms": [
                {
                    "AlarmName": "TargetTracking-my-asg-
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196"
                },
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-
a010-c1aaa35da296",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-a010-
c1aaa35da296"
                }
            ],
            "TargetTrackingConfiguration": {
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "ALBRequestCountPerTarget",
                    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-
alb-target-group/943f017f100becff"
```

```
                },
                "TargetValue": 1000.0,
                "DisableScaleIn": false
            },
            "Enabled": true
        },
        {
            "AutoScalingGroupName": "my-asg",
            "PolicyName": "cpu40-target-tracking-scaling-policy",
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:5fd26f71-39d4-4690-82a9-
b8515c45cdde:autoScalingGroupName/my-asg:policyName/cpu40-target-tracking-scaling-
policy",
            "PolicyType": "TargetTrackingScaling",
            "StepAdjustments": [],
            "Alarms": [
                {
                    "AlarmName": "TargetTracking-my-asg-
AlarmHigh-139f9789-37b9-42ad-bea5-b5b147d7f473",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-139f9789-37b9-42ad-bea5-
b5b147d7f473"
                },
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmLow-bd681c67-
fc18-4c56-8468-fb8e413009c9",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-bd681c67-fc18-4c56-8468-
fb8e413009c9"
                }
            ],
            "TargetTrackingConfiguration": {
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "ASGAverageCPUUtilization"
                },
                "TargetValue": 40.0,
                "DisableScaleIn": false
            },
            "Enabled": true
        }
    ]
}
```

Pour plus d'informations, consultez Dynamic Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour décrire les politiques de dimensionnement d'un nom spécifié

Pour renvoyer des politiques de dimensionnement spécifiques, utilisez l'`--policy-names`option.

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg \
    --policy-names cpu40-target-tracking-scaling-policy
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Dynamic Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 3 : pour décrire un certain nombre de politiques de dimensionnement

Pour renvoyer un nombre spécifique de politiques, utilisez l'`--max-items`option.

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg \
    --max-items 1
```

Voir l'exemple 1 pour un exemple de sortie.

Si la sortie inclut un `NextToken` champ, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel ultérieur pour obtenir les politiques supplémentaires.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-
token Z3M3LMPEXAMPLE
```

Pour plus d'informations, consultez Dynamic Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribePoliciesà la section Référence des AWS CLI commandes.

`describe-scaling-activities`

L'exemple de code suivant montre comment utiliser`describe-scaling-activities`.

AWS CLI

Exemple 1 : pour décrire les activités de dimensionnement pour le groupe spécifié

Cet exemple décrit les activités de dimensionnement pour le groupe Auto Scaling spécifié.

```
aws autoscaling describe-scaling-activities \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",
            "Description": "Launching a new EC2 instance: i-0d44425630326060f",
            "AutoScalingGroupName": "my-asg",
            "Cause": "At 2020-10-30T19:35:51Z a user request update of
 AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired
 capacity from 0 to 16.  At 2020-10-30T19:36:07Z an instance was started in response
 to a difference between desired and actual capacity, increasing the capacity from 0
 to 16.",
            "StartTime": "2020-10-30T19:36:09.766Z",
            "EndTime": "2020-10-30T19:36:41Z",
            "StatusCode": "Successful",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":
\"us-west-2b\"}"
        }
    ]
}
```

Pour plus d'informations, consultez Vérifier une activité de dimensionnement pour un groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour décrire les activités de dimensionnement pour un groupe supprimé

Pour décrire les activités de dimensionnement après la suppression du groupe Auto Scaling, ajoutez l'`--include-deleted-groups`option.

```
aws autoscaling describe-scaling-activities \
    --auto-scaling-group-name my-asg \
```

```
    --include-deleted-groups
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
            "Description": "Launching a new EC2 instance.  Status Reason: Your Spot
 request price of 0.001 is lower than the minimum required Spot request fulfillment
 price of 0.0031. Launching EC2 instance failed.",
            "AutoScalingGroupName": "my-asg",
            "Cause": "At 2021-01-13T20:47:24Z a user request update of
 AutoScalingGroup constraints to min: 1, max: 5, desired: 3 changing the desired
 capacity from 0 to 3.  At 2021-01-13T20:47:27Z an instance was started in response
 to a difference between desired and actual capacity, increasing the capacity from 0
 to 3.",
            "StartTime": "2021-01-13T20:47:30.094Z",
            "EndTime": "2021-01-13T20:47:30Z",
            "StatusCode": "Failed",
            "StatusMessage": "Your Spot request price of 0.001 is lower than the
 minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance
 failed.",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":
\"us-west-2b\"}",
            "AutoScalingGroupState": "Deleted",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
        }
    ]
}
```

Pour plus d'informations, consultez la section Résoudre les problèmes liés à Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 3 : pour décrire un nombre spécifié d'activités de mise à l'échelle

Pour renvoyer un certain nombre d'activités, utilisez l'`--max-items`option.

```
aws autoscaling describe-scaling-activities \
```

```
    --max-items 1
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",
            "Description": "Launching a new EC2 instance: i-0d44425630326060f",
            "AutoScalingGroupName": "my-asg",
            "Cause": "At 2020-10-30T19:35:51Z a user request update of
 AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired
 capacity from 0 to 16.  At 2020-10-30T19:36:07Z an instance was started in response
 to a difference between desired and actual capacity, increasing the capacity from 0
 to 16.",
            "StartTime": "2020-10-30T19:36:09.766Z",
            "EndTime": "2020-10-30T19:36:41Z",
            "StatusCode": "Successful",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":
\"us-west-2b\"}"
        }
    ]
}
```

Si la sortie inclut un `NextToken` champ, il y a d'autres activités. Pour obtenir les activités supplémentaires, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel suivant comme suit.

```
aws autoscaling describe-scaling-activities \
    --starting-token Z3M3LMPEXAMPLE
```

Pour plus d'informations, consultez Vérifier une activité de dimensionnement pour un groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalingActivitiesà la section Référence des AWS CLI commandes.

`describe-scaling-process-types`

L'exemple de code suivant montre comment utiliser`describe-scaling-process-types`.

AWS CLI

Pour décrire les types de processus disponibles

Cet exemple décrit les types de processus disponibles.

```
aws autoscaling describe-scaling-process-types
```

Sortie :

```
{
    "Processes": [
        {
            "ProcessName": "AZRebalance"
        },
        {
            "ProcessName": "AddToLoadBalancer"
        },
        {
            "ProcessName": "AlarmNotification"
        },
        {
            "ProcessName": "HealthCheck"
        },
        {
            "ProcessName": "InstanceRefresh"
        },
        {
            "ProcessName": "Launch"
        },
        {
            "ProcessName": "ReplaceUnhealthy"
        },
        {
            "ProcessName": "ScheduledActions"
        },
        {
            "ProcessName": "Terminate"
        }
    ]
}
```

Pour plus d'informations, consultez la section Suspendre et reprise des processus de dimensionnement dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalingProcessTypesà la section Référence des AWS CLI commandes.

`describe-scheduled-actions`

L'exemple de code suivant montre comment utiliser`describe-scheduled-actions`.

AWS CLI

Exemple 1 : pour décrire toutes les actions planifiées

Cet exemple décrit toutes vos actions planifiées.

```
aws autoscaling describe-scheduled-actions
```

Sortie :

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

Pour plus d'informations, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour décrire les actions planifiées pour le groupe spécifié

Pour décrire les actions planifiées pour un groupe Auto Scaling spécifique, utilisez l'`--auto-scaling-group-name`option.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

Pour plus d'informations, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 3 : pour décrire l'action planifiée spécifiée

Pour décrire une action planifiée spécifique, utilisez l'`--scheduled-action-names`option.

```
aws autoscaling describe-scheduled-actions \
    --scheduled-action-names my-recurring-action
```

Sortie :

```
{
```

```
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

Pour plus d'informations, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 4 : Pour décrire des actions planifiées avec une heure de début spécifiée

Pour décrire les actions planifiées qui démarrent à une heure précise, utilisez l'`--start-time`option.

```
aws autoscaling describe-scheduled-actions \
    --start-time "2023-12-01T04:00:00Z"
```

Sortie :

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
```

```
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

Pour plus d'informations, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 5 : Pour décrire les actions planifiées qui se terminent à une heure spécifiée

Pour décrire les actions planifiées qui se terminent à un moment précis, utilisez l'`--end-time`option.

```
aws autoscaling describe-scheduled-actions \
    --end-time "2023-12-01T04:00:00Z"
```

Sortie :

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

Pour plus d'informations, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 6 : Pour décrire un nombre spécifié d'actions planifiées

Pour renvoyer un nombre spécifique d'actions planifiées, utilisez l'`--max-items`option.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --max-items 1
```

Sortie :

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

Si la sortie inclut un `NextToken` champ, d'autres actions sont planifiées. Pour obtenir les actions planifiées supplémentaires, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel suivant, comme suit.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

Pour plus d'informations, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledActionsà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire tous les tags

Cet exemple décrit tous vos tags.

```
aws autoscaling describe-tags
```

Sortie :

```
{
    "Tags": [
        {
            "ResourceType": "auto-scaling-group",
            "ResourceId": "my-asg",
            "PropagateAtLaunch": true,
            "Value": "Research",
            "Key": "Dept"
        },
        {
            "ResourceType": "auto-scaling-group",
            "ResourceId": "my-asg",
            "PropagateAtLaunch": true,
            "Value": "WebServer",
            "Key": "Role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Tagging Auto Scaling groups and instances dans le Amazon EC2 Auto Scaling User Guide.

Exemple 2 : pour décrire les balises d'un groupe spécifique

Pour décrire les balises d'un groupe Auto Scaling spécifique, utilisez l'`--filters`option.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

Pour plus d'informations, consultez la section Tagging Auto Scaling groups and instances dans le Amazon EC2 Auto Scaling User Guide.

Exemple 3 : pour décrire le nombre de balises spécifié

Pour renvoyer un nombre spécifique de balises, utilisez l'`--max-items`option.

```
aws autoscaling describe-tags \
    --max-items 1
```

Si la sortie inclut un `NextToken` champ, il y a d'autres balises. Pour obtenir les balises supplémentaires, utilisez la valeur de ce champ avec l'`--starting-token`option lors d'un appel suivant comme suit.

```
aws autoscaling describe-tags \
    --filters Name=auto-scaling-group,Values=my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

Pour plus d'informations, consultez la section Tagging Auto Scaling groups and instances dans le Amazon EC2 Auto Scaling User Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`describe-termination-policy-types`

L'exemple de code suivant montre comment utiliser`describe-termination-policy-types`.

AWS CLI

Pour décrire les types de politiques de résiliation disponibles

Cet exemple décrit les types de politiques de résiliation disponibles.

```
aws autoscaling describe-termination-policy-types
```

Sortie :

```
{
```

```
    "TerminationPolicyTypes": [
        "AllocationStrategy",
        "ClosestToNextInstanceHour",
        "Default",
        "NewestInstance",
        "OldestInstance",
        "OldestLaunchConfiguration",
        "OldestLaunchTemplate"
    ]
}
```

Pour plus d'informations, consultez la section Contrôler quelles instances Auto Scaling se terminent pendant le dimensionnement dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeTerminationPolicyTypesà la section Référence des AWS CLI commandes.

`describe-warm-pool`

L'exemple de code suivant montre comment utiliser`describe-warm-pool`.

AWS CLI

Pour décrire une piscine chaude

Cet exemple décrit le pool de chaleur pour le groupe Auto Scaling spécifié.

```
aws autoscaling describe-warm-pool \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "WarmPoolConfiguration": {
        "MinSize": 2,
        "PoolState": "Stopped"
    },
    "Instances": [
        {
            "InstanceId": "i-070a5bbc7e7f40dc5",
            "InstanceType": "t2.micro",
```

```
            "AvailabilityZone": "us-west-2c",
            "LifecycleState": "Warmed:Pending",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-00a731f6e9fa48610",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "6"
            }
        },
        {
            "InstanceId": "i-0b52f061814d3bd2d",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2b",
            "LifecycleState": "Warmed:Pending",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-00a731f6e9fa48610",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "6"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Warm pools for Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeWarmPoolà la section Référence des AWS CLI commandes.

`detach-instances`

L'exemple de code suivant montre comment utiliser`detach-instances`.

AWS CLI

Pour détacher une instance d'un groupe Auto Scaling

Cet exemple détache l'instance spécifiée du groupe Auto Scaling spécifié.

```
aws autoscaling detach-instances \
    --instance-ids i-030017cfa84b20135 \
    --auto-scaling-group-name my-asg \
```

```
    --should-decrement-desired-capacity
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "5091cb52-547a-47ce-a236-c9ccbc2cb2c9",
            "AutoScalingGroupName": "my-asg",
            "Description": "Detaching EC2 instance: i-030017cfa84b20135",
            "Cause": "At 2020-10-31T17:35:04Z instance i-030017cfa84b20135 was
 detached in response to a user request, shrinking the capacity from 2 to 1.",
            "StartTime": "2020-04-12T15:02:16.179Z",
            "StatusCode": "InProgress",
            "Progress": 50,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DetachInstancesà la section Référence des AWS CLI commandes.

`detach-load-balancer-target-groups`

L'exemple de code suivant montre comment utiliser`detach-load-balancer-target-groups`.

AWS CLI

Pour détacher un groupe cible d'équilibreurs de charge d'un groupe Auto Scaling

Cet exemple détache le groupe cible d'équilibreurs de charge spécifié du groupe Auto Scaling spécifié.

```
aws autoscaling detach-load-balancer-target-groups \
    --auto-scaling-group-name my-asg \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Cette commande ne produit aucune sortie

Pour plus d'informations, consultez la section Attacher un équilibreur de charge à votre groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DetachLoadBalancerTargetGroupsà la section Référence des AWS CLI commandes.

`detach-load-balancers`

L'exemple de code suivant montre comment utiliser`detach-load-balancers`.

AWS CLI

Pour détacher un Classic Load Balancer d'un groupe Auto Scaling

Cet exemple détache le Classic Load Balancer spécifié du groupe Auto Scaling spécifié.

```
aws autoscaling detach-load-balancers \
    --load-balancer-names my-load-balancer \
    --auto-scaling-group-name my-asg
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Attacher un équilibreur de charge à votre groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DetachLoadBalancersà la section Référence des AWS CLI commandes.

`disable-metrics-collection`

L'exemple de code suivant montre comment utiliser`disable-metrics-collection`.

AWS CLI

Pour désactiver la collecte de métriques pour un groupe Auto Scaling

Cet exemple désactive la collecte de la `GroupDesiredCapacity` métrique pour le groupe Auto Scaling spécifié.

```
aws autoscaling disable-metrics-collection \
    --auto-scaling-group-name my-asg \
```

```
    --metrics GroupDesiredCapacity
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance CloudWatch des métriques pour vos groupes et instances Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DisableMetricsCollectionà la section Référence des AWS CLI commandes.

`enable-metrics-collection`

L'exemple de code suivant montre comment utiliser`enable-metrics-collection`.

AWS CLI

Exemple 1 : Pour activer la collecte de métriques pour un groupe Auto Scaling

Cet exemple active la collecte de données pour le groupe Auto Scaling spécifié.

```
aws autoscaling enable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --granularity "1Minute"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance CloudWatch des métriques pour vos groupes et instances Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : Pour collecter des données pour la métrique spécifiée pour un groupe Auto Scaling

Pour collecter des données pour une métrique spécifique, utilisez l'`--metrics`option.

```
aws autoscaling enable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --metrics GroupDesiredCapacity --granularity "1Minute"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance CloudWatch des métriques pour vos groupes et instances Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous EnableMetricsCollectionà la section Référence des AWS CLI commandes.

`enter-standby`

L'exemple de code suivant montre comment utiliser`enter-standby`.

AWS CLI

Pour déplacer des instances en mode veille

Cet exemple met l'instance spécifiée en mode veille. Cela est utile pour mettre à jour ou dépanner une instance actuellement en service.

```
aws autoscaling enter-standby \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg \
    --should-decrement-desired-capacity
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "ffa056b4-6ed3-41ba-ae7c-249dfae6eba1",
            "AutoScalingGroupName": "my-asg",
            "Description": "Moving EC2 instance to Standby: i-061c63c5eb45f0416",
            "Cause": "At 2020-10-31T20:31:00Z instance i-061c63c5eb45f0416 was moved
 to standby in response to a user request, shrinking the capacity from 1 to 0.",
            "StartTime": "2020-10-31T20:31:00.949Z",
            "StatusCode": "InProgress",
            "Progress": 50,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
}
```

Pour plus d'informations, consultez le cycle de vie des instances Amazon EC2 Auto Scaling dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous EnterStandbyà la section Référence des AWS CLI commandes.

`execute-policy`

L'exemple de code suivant montre comment utiliser`execute-policy`.

AWS CLI

Pour exécuter une politique de dimensionnement

Cet exemple exécute la politique de dimensionnement nommée `my-step-scale-out-policy` pour le groupe Auto Scaling spécifié.

```
aws autoscaling execute-policy \
    --auto-scaling-group-name my-asg \
    --policy-name  my-step-scale-out-policy \
    --metric-value 95 \
    --breach-threshold 80
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Step and Simple Scaling policies dans le manuel Amazon EC2 Auto Scaling User Guide.

- Pour plus de détails sur l'API, reportez-vous ExecutePolicyà la section Référence des AWS CLI commandes.

`exit-standby`

L'exemple de code suivant montre comment utiliser`exit-standby`.

AWS CLI

Pour faire sortir les instances du mode veille

Cet exemple fait sortir l'instance spécifiée du mode veille.

```
aws autoscaling exit-standby \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "142928e1-a2dc-453a-9b24-b85ad6735928",
            "AutoScalingGroupName": "my-asg",
            "Description": "Moving EC2 instance out of Standby:
 i-061c63c5eb45f0416",
            "Cause": "At 2020-10-31T20:32:50Z instance i-061c63c5eb45f0416 was moved
 out of standby in response to a user request, increasing the capacity from 0 to
 1.",
            "StartTime": "2020-10-31T20:32:50.222Z",
            "StatusCode": "PreInService",
            "Progress": 30,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
}
```

Pour plus d'informations, consultez la section Suppression temporaire d'instances de votre groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous ExitStandbyà la section Référence des AWS CLI commandes.

`put-lifecycle-hook`

L'exemple de code suivant montre comment utiliser`put-lifecycle-hook`.

AWS CLI

Exemple 1 : pour créer un hook de cycle de vie

Cet exemple crée un hook de cycle de vie qui sera invoqué sur toutes les instances nouvellement lancées, avec un délai d'expiration de 4 800 secondes. Cela est utile pour maintenir les instances en état d'attente jusqu'à ce que les scripts de données utilisateur soient terminés, ou pour appeler une fonction AWS Lambda à l'aide de. EventBridge

```
aws autoscaling put-lifecycle-hook \
    --auto-scaling-group-name my-asg \
```

```
    --lifecycle-hook-name my-launch-hook \
    --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \
    --heartbeat-timeout 4800
```

Cette commande ne produit aucun résultat. Si un hook de cycle de vie portant le même nom existe déjà, il sera remplacé par le nouveau hook de cycle de vie.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour envoyer un e-mail Amazon SNS pour vous informer des transitions d'état d'instance

Cet exemple crée un hook de cycle de vie avec la rubrique Amazon SNS et le rôle IAM à utiliser pour recevoir des notifications lors du lancement de l'instance.

```
aws autoscaling put-lifecycle-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-hook-name my-launch-hook \
    --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \
    --notification-target-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \
    --role-arn arn:aws:iam::123456789012:role/my-auto-scaling-role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 3 : pour publier un message dans une file d'attente Amazon SQS

Cet exemple crée un hook de cycle de vie qui publie un message contenant des métadonnées dans la file d'attente Amazon SQS spécifiée.

```
aws autoscaling put-lifecycle-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-hook-name my-launch-hook \
    --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \
    --notification-target-arn arn:aws:sqs:us-west-2:123456789012:my-sqs-queue \
    --role-arn arn:aws:iam::123456789012:role/my-notification-role \
    --notification-metadata "SQS message metadata"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutLifecycleHookà la section Référence des AWS CLI commandes.

`put-notification-configuration`

L'exemple de code suivant montre comment utiliser`put-notification-configuration`.

AWS CLI

Pour ajouter une notification

Cet exemple ajoute la notification spécifiée au groupe Auto Scaling spécifié.

```
aws autoscaling put-notification-configuration \
    --auto-scaling-group-name my-asg \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \
    --notification-type autoscaling:TEST_NOTIFICATION
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Recevoir des notifications Amazon SNS lorsque votre groupe Auto Scaling évolue dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutNotificationConfigurationà la section Référence des AWS CLI commandes.

`put-scaling-policy`

L'exemple de code suivant montre comment utiliser`put-scaling-policy`.

AWS CLI

Pour ajouter une politique de dimensionnement du suivi des cibles à un groupe Auto Scaling

L'`put-scaling-policy`exemple suivant applique une politique de dimensionnement du suivi des cibles au groupe Auto Scaling spécifié. La sortie contient les noms ARNs et les noms des deux CloudWatch alarmes créées en votre nom. Si une stratégie de dimensionnement portant le même nom existe déjà, elle sera remplacée par la nouvelle stratégie de dimensionnement.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \
  --policy-name alb1000-target-tracking-scaling-policy \
  --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
```

Contenu de `config.json` :

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
          "PredefinedMetricType": "ALBRequestCountPerTarget",
          "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
     }
}
```

Sortie :

```
{
     "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:228f02c2-
c665-4bfd-aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/alb1000-target-
tracking-scaling-policy",
     "Alarms": [
         {
             "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
             "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
         },
         {
             "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
             "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
         }
     ]
 }
```

Pour plus d'exemples, consultez la section Exemples de politiques de dimensionnement pour l'interface de ligne de AWS commande (AWS CLI) dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutScalingPolicyà la section Référence des AWS CLI commandes.

`put-scheduled-update-group-action`

L'exemple de code suivant montre comment utiliser`put-scheduled-update-group-action`.

AWS CLI

Exemple 1 : Pour ajouter une action planifiée à un groupe Auto Scaling

Cet exemple ajoute l'action planifiée spécifiée au groupe Auto Scaling spécifié.

```
aws autoscaling put-scheduled-update-group-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-scheduled-action \
    --start-time "2023-05-12T08:00:00Z" \
    --min-size 2 \
    --max-size 6 \
    --desired-capacity 4
```

Cette commande ne produit aucun résultat. Si une action planifiée portant le même nom existe déjà, elle sera remplacée par la nouvelle action planifiée.

Pour plus d'exemples, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour définir un calendrier récurrent

Cet exemple crée une action planifiée pour s'adapter à un calendrier récurrent dont l'exécution est prévue à 00h30 les 1er janvier, juin et décembre de chaque année.

```
aws autoscaling put-scheduled-update-group-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-recurring-action \
    --recurrence "30 0 1 1,6,12 *" \
    --min-size 2 \
    --max-size 6 \
    --desired-capacity 4
```

Cette commande ne produit aucun résultat. Si une action planifiée portant le même nom existe déjà, elle sera remplacée par la nouvelle action planifiée.

Pour plus d'exemples, consultez Scheduled Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutScheduledUpdateGroupActionà la section Référence des AWS CLI commandes.

`put-warm-pool`

L'exemple de code suivant montre comment utiliser`put-warm-pool`.

AWS CLI

Pour créer une piscine chaude

L'exemple suivant crée un pool de chaleur pour le groupe Auto Scaling spécifié.

```
aws autoscaling put-warm-pool \
    --auto-scaling-group-name my-asg \
    --min-size 2
```

Cette commande ne produit aucun résultat. Si une piscine chaude existe déjà, elle sera mise à jour.

Pour plus d'informations, consultez Warm pools for Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous PutWarmPoolà la section Référence des AWS CLI commandes.

`record-lifecycle-action-heartbeat`

L'exemple de code suivant montre comment utiliser`record-lifecycle-action-heartbeat`.

AWS CLI

Pour enregistrer le rythme cardiaque d'une action pendant le cycle de vie

Cet exemple enregistre le rythme cardiaque d'une action du cycle de vie afin de maintenir l'instance en attente.

```
aws autoscaling record-lifecycle-action-heartbeat \
```

```
    --lifecycle-hook-name my-launch-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les hooks du cycle de vie d'Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous RecordLifecycleActionHeartbeatà la section Référence des AWS CLI commandes.

`resume-processes`

L'exemple de code suivant montre comment utiliser`resume-processes`.

AWS CLI

Pour reprendre les processus suspendus

Cet exemple reprend le processus de dimensionnement suspendu spécifié pour le groupe Auto Scaling spécifié.

```
aws autoscaling resume-processes \
    --auto-scaling-group-name my-asg \
    --scaling-processes AlarmNotification
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suspendre et reprise des processus de dimensionnement dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous ResumeProcessesà la section Référence des AWS CLI commandes.

`rollback-instance-refresh`

L'exemple de code suivant montre comment utiliser`rollback-instance-refresh`.

AWS CLI

Pour annuler l'actualisation d'une instance

L'`rollback-instance-refresh`exemple suivant annule une actualisation d'instance en cours pour le groupe Auto Scaling spécifié.

```
aws autoscaling rollback-instance-refresh \
    --auto-scaling-group-name my-asg
```

Sortie :

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Pour plus d'informations, consultez la section Annuler les modifications par une annulation dans le manuel Amazon EC2 Auto Scaling User Guide.

- Pour plus de détails sur l'API, reportez-vous RollbackInstanceRefreshà la section Référence des AWS CLI commandes.

`set-desired-capacity`

L'exemple de code suivant montre comment utiliser`set-desired-capacity`.

AWS CLI

Pour définir la capacité souhaitée pour un groupe Auto Scaling

Cet exemple définit la capacité souhaitée pour le groupe Auto Scaling spécifié.

```
aws autoscaling set-desired-capacity \
    --auto-scaling-group-name my-asg \
    --desired-capacity 2 \
    --honor-cooldown
```

Cette commande revient à l'invite en cas de succès.

- Pour plus de détails sur l'API, reportez-vous SetDesiredCapacityà la section Référence des AWS CLI commandes.

`set-instance-health`

L'exemple de code suivant montre comment utiliser`set-instance-health`.

AWS CLI

Pour définir l'état de santé d'une instance

Cet exemple définit l'état de santé de l'instance spécifiée sur`Unhealthy`.

```
aws autoscaling set-instance-health \
    --instance-id i-061c63c5eb45f0416 \
    --health-status Unhealthy
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetInstanceHealthà la section Référence des AWS CLI commandes.

`set-instance-protection`

L'exemple de code suivant montre comment utiliser`set-instance-protection`.

AWS CLI

Exemple 1 : pour activer le paramètre de protection d'instance pour une instance

Cet exemple active la protection de l'instance pour l'instance spécifiée.

```
aws autoscaling set-instance-protection \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg --protected-from-scale-in
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour désactiver le paramètre de protection d'instance pour une instance

Cet exemple désactive la protection de l'instance pour l'instance spécifiée.

```
aws autoscaling set-instance-protection \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg \
    --no-protected-from-scale-in
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetInstanceProtectionà la section Référence des AWS CLI commandes.

`start-instance-refresh`

L'exemple de code suivant montre comment utiliser`start-instance-refresh`.

AWS CLI

Exemple 1 : pour démarrer l'actualisation d'une instance à l'aide des paramètres de ligne de commande

L'`start-instance-refresh`exemple suivant lance une actualisation d'instance à l'aide d'arguments de ligne de commande. Le `preferences` paramètre facultatif spécifie un `InstanceWarmup` nombre de `60` secondes et un `MinHealthyPercentage` `50` pourcentage.

```
aws autoscaling start-instance-refresh \
    --auto-scaling-group-name my-asg \
    --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```

Sortie :

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Pour plus d'informations, consultez Démarrer l'actualisation d'une instance dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

Exemple 2 : pour démarrer l'actualisation d'une instance à l'aide d'un fichier JSON

L'`start-instance-refresh`exemple suivant lance une actualisation d'instance à l'aide d'un fichier JSON. Vous pouvez spécifier le groupe Auto Scaling et définir la configuration et les préférences souhaitées dans un fichier JSON, comme illustré dans l'exemple suivant.

```
aws autoscaling start-instance-refresh \
    --cli-input-json file://config.json
```

Contenu de `config.json` :

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
        "LaunchTemplate": {
            "LaunchTemplateId": "lt-068f72b729example",
            "Version": "$Default"
        }
    },
    "Preferences": {
        "InstanceWarmup": 60,
        "MinHealthyPercentage": 50,
        "AutoRollback": true,
        "ScaleInProtectedInstances": Ignore,
        "StandbyInstances": Terminate
    }
}
```

Sortie :

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Pour plus d'informations, consultez Démarrer l'actualisation d'une instance dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous StartInstanceRefreshà la section Référence des AWS CLI commandes.

`suspend-processes`

L'exemple de code suivant montre comment utiliser`suspend-processes`.

AWS CLI

Pour suspendre les processus Auto Scaling

Cet exemple suspend le processus de dimensionnement spécifié pour le groupe Auto Scaling spécifié.

```
aws autoscaling suspend-processes \
    --auto-scaling-group-name my-asg \
```

```
    --scaling-processes AlarmNotification
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suspendre et reprise des processus de dimensionnement dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous SuspendProcessesà la section Référence des AWS CLI commandes.

`terminate-instance-in-auto-scaling-group`

L'exemple de code suivant montre comment utiliser`terminate-instance-in-auto-scaling-group`.

AWS CLI

Pour mettre fin à une instance d'un groupe Auto Scaling

Cet exemple met fin à l'instance spécifiée à partir du groupe Auto Scaling spécifié sans mettre à jour la taille du groupe. Amazon EC2 Auto Scaling lance une instance de remplacement après la fin de l'instance spécifiée.

```
aws autoscaling terminate-instance-in-auto-scaling-group \
    --instance-id i-061c63c5eb45f0416 \
    --no-should-decrement-desired-capacity
```

Sortie :

```
{
    "Activities": [
        {
            "ActivityId": "8c35d601-793c-400c-fcd0-f64a27530df7",
            "AutoScalingGroupName": "my-asg",
            "Description": "Terminating EC2 instance: i-061c63c5eb45f0416",
            "Cause": "",
            "StartTime": "2020-10-31T20:34:25.680Z",
            "StatusCode": "InProgress",
            "Progress": 0,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous TerminateInstanceInAutoScalingGroupà la section Référence des AWS CLI commandes.

`update-auto-scaling-group`

L'exemple de code suivant montre comment utiliser`update-auto-scaling-group`.

AWS CLI

Exemple 1 : Pour mettre à jour les limites de taille d'un groupe Auto Scaling

Cet exemple met à jour le groupe Auto Scaling spécifié avec une taille minimale de 2 et une taille maximale de 10.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --min-size 2 \
    --max-size 10
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition des limites de capacité pour votre groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 2 : pour ajouter des contrôles de santé d'Elastic Load Balancing et spécifier les zones de disponibilité et les sous-réseaux à utiliser

Cet exemple met à jour le groupe Auto Scaling spécifié pour ajouter les tests de santé d'Elastic Load Balancing. Cette commande met également à jour la valeur de `--vpc-zone-identifier` avec une liste de sous-réseaux IDs dans plusieurs zones de disponibilité.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --health-check-type ELB \
    --health-check-grace-period 600 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Elastic Load Balancing et Amazon EC2 Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 3 : pour mettre à jour le groupe de placement et la politique de résiliation

Cet exemple met à jour le groupe de placement et la politique de résiliation à utiliser.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --placement-group my-placement-group \
    --termination-policies "OldestInstance"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 4 : Pour utiliser la dernière version du modèle de lancement

Cet exemple met à jour le groupe Auto Scaling spécifié pour utiliser la dernière version du modèle de lancement spécifié.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modèles de lancement dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 5 : Pour utiliser une version spécifique du modèle de lancement

Cet exemple met à jour le groupe Auto Scaling spécifié pour utiliser une version spécifique d'un modèle de lancement au lieu de la version la plus récente ou de la version par défaut.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modèles de lancement dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Exemple 6 : pour définir une politique d'instances mixtes et activer le rééquilibrage des capacités

Cet exemple met à jour le groupe Auto Scaling spécifié pour utiliser une politique d'instances mixtes et permet le rééquilibrage des capacités. Cette structure vous permet de spécifier des groupes dotés de capacités ponctuelles et à la demande et d'utiliser différents modèles de lancement pour différentes architectures.

```
aws autoscaling update-auto-scaling-group \
    --cli-input-json file://~/config.json
```

Contenu de `config.json` :

```
{
    "AutoScalingGroupName": "my-asg",
    "CapacityRebalance": true,
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template-for-x86",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c6g.large",
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template-for-arm",
                        "Version": "$Latest"
                    }
                },
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
```

```
            "SpotAllocationStrategy": "capacity-optimized"
        }
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes Auto Scaling avec plusieurs types d'instances et options d'achat dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous UpdateAutoScalingGroupà la section Référence des AWS CLI commandes.

Exemples d'Auto Scaling Plans utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Auto Scaling Plans.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-scaling-plan`

L'exemple de code suivant montre comment utiliser`create-scaling-plan`.

AWS CLI

Pour créer un plan de mise à l'échelle

L'`create-scaling-plan`exemple suivant crée un plan de dimensionnement nommé à `my-scaling-plan` l'aide d'un fichier JSON déjà créé (nommé config.json). La structure du plan de dimensionnement inclut une instruction de dimensionnement pour un groupe Auto Scaling

nommémy-asg. Elle spécifie la propriété TagFilters en tant que source de l'application et active le dimensionnement prédictif et dynamique.

```
aws autoscaling-plans create-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --cli-input-json file://~/config.json
```

Contenu du config.json fichier :

```
{
    "ApplicationSource": {
        "TagFilters": [
            {
                "Key": "purpose",
                "Values": [
                    "my-application"
                ]
            }
        ]
    },
    "ScalingInstructions": [
        {
            "ServiceNamespace": "autoscaling",
            "ResourceId": "autoScalingGroup/my-asg",
            "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",
            "ScheduledActionBufferTime": 300,
            "PredictiveScalingMaxCapacityBehavior":
 "SetForecastCapacityToMaxCapacity",
            "PredictiveScalingMode": "ForecastAndScale",
            "PredefinedLoadMetricSpecification": {
                "PredefinedLoadMetricType": "ASGTotalCPUUtilization"
            },
            "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
            "MinCapacity": 1,
            "MaxCapacity": 4,
            "TargetTrackingConfigurations": [
                {
                    "PredefinedScalingMetricSpecification": {
                        "PredefinedScalingMetricType": "ASGAverageCPUUtilization"
                    },
                    "TargetValue": 50
                }
            ]
```

```
        }
    ]
}
```

Sortie :

```
{
"ScalingPlanVersion": 1
}
```

Pour plus d'informations, consultez le guide de l'utilisateur d'AWS Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous CreateScalingPlanà la section Référence des AWS CLI commandes.

`delete-scaling-plan`

L'exemple de code suivant montre comment utiliser`delete-scaling-plan`.

AWS CLI

Pour supprimer un plan de dimensionnement

L'`delete-scaling-plan`exemple suivant supprime le plan de dimensionnement spécifié.

```
aws autoscaling-plans delete-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez le guide de l'utilisateur d'AWS Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DeleteScalingPlanà la section Référence des AWS CLI commandes.

`describe-scaling-plan-resources`

L'exemple de code suivant montre comment utiliser`describe-scaling-plan-resources`.

AWS CLI

Pour décrire les ressources évolutives d'un plan de mise à l'échelle

L'`describe-scaling-plan-resources`exemple suivant affiche les détails de la ressource évolutive unique (un groupe Auto Scaling) associée au plan de dimensionnement spécifié.

```
aws autoscaling-plans describe-scaling-plan-resources \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1
```

Sortie :

```
{
    "ScalingPlanResources": [
        {
            "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",
            "ScalingPlanVersion": 1,
            "ResourceId": "autoScalingGroup/my-asg",
            "ScalingStatusCode": "Active",
            "ScalingStatusMessage": "Target tracking scaling policies have been
 applied to the resource.",
            "ScalingPolicies": [
                {
                    "PolicyName": "AutoScaling-my-asg-b1ab65ae-4be3-4634-bd64-
c7471662b251",
                    "PolicyType": "TargetTrackingScaling",
                    "TargetTrackingConfiguration": {
                        "PredefinedScalingMetricSpecification": {
                            "PredefinedScalingMetricType":
 "ALBRequestCountPerTarget",
                            "ResourceLabel": "app/my-alb/f37c06a68c1748aa/
targetgroup/my-target-group/6d4ea56ca2d6a18d"
                        },
                        "TargetValue": 40.0
                    }
                }
            ],
            "ServiceNamespace": "autoscaling",
            "ScalingPlanName": "my-scaling-plan"
        }
    ]
}
```

Pour plus d'informations, consultez What Is AWS Auto Scaling ? dans le Guide de l'utilisateur d'AWS Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalingPlanResourcesà la section Référence des AWS CLI commandes.

`describe-scaling-plans`

L'exemple de code suivant montre comment utiliser`describe-scaling-plans`.

AWS CLI

Pour décrire un plan de mise à l'échelle

L'`describe-scaling-plans`exemple suivant affiche les détails du plan de dimensionnement spécifié.

```
aws autoscaling-plans describe-scaling-plans \
    --scaling-plan-names scaling-plan-with-asg-and-ddb
```

Sortie :

```
{
    "ScalingPlans": [
        {
            "LastMutatingRequestTime": 1565388443.963,
            "ScalingPlanVersion": 1,
            "CreationTime": 1565388443.963,
            "ScalingInstructions": [
                {
                    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
                    "ScalableDimension":
 "autoscaling:autoScalingGroup:DesiredCapacity",
                    "TargetTrackingConfigurations": [
                        {
                            "PredefinedScalingMetricSpecification": {
                                "PredefinedScalingMetricType":
 "ASGAverageCPUUtilization"
                            },
                            "TargetValue": 50.0,
                            "EstimatedInstanceWarmup": 300,
                            "DisableScaleIn": false
                        }
                    ],
                    "ResourceId": "autoScalingGroup/my-asg",
```

```
                    "DisableDynamicScaling": false,
                    "MinCapacity": 1,
                    "ServiceNamespace": "autoscaling",
                    "MaxCapacity": 10
                },
                {
                    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
                    "ScalableDimension": "dynamodb:table:ReadCapacityUnits",
                    "TargetTrackingConfigurations": [
                        {
                            "PredefinedScalingMetricSpecification": {
                                "PredefinedScalingMetricType":
"DynamoDBReadCapacityUtilization"
                            },
                            "TargetValue": 50.0,
                            "ScaleInCooldown": 60,
                            "DisableScaleIn": false,
                            "ScaleOutCooldown": 60
                        }
                    ],
                    "ResourceId": "table/my-table",
                    "DisableDynamicScaling": false,
                    "MinCapacity": 5,
                    "ServiceNamespace": "dynamodb",
                    "MaxCapacity": 10000
                },
                {
                    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
                    "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
                    "TargetTrackingConfigurations": [
                        {
                            "PredefinedScalingMetricSpecification": {
                                "PredefinedScalingMetricType":
"DynamoDBWriteCapacityUtilization"
                            },
                            "TargetValue": 50.0,
                            "ScaleInCooldown": 60,
                            "DisableScaleIn": false,
                            "ScaleOutCooldown": 60
                        }
                    ],
                    "ResourceId": "table/my-table",
                    "DisableDynamicScaling": false,
                    "MinCapacity": 5,
```

```
                    "ServiceNamespace": "dynamodb",
                    "MaxCapacity": 10000
                }
            ],
            "ApplicationSource": {
                "TagFilters": [
                    {
                        "Values": [
                            "my-application-id"
                        ],
                        "Key": "application"
                    }
                ]
            },
            "StatusStartTime": 1565388455.836,
            "ScalingPlanName": "scaling-plan-with-asg-and-ddb",
            "StatusMessage": "Scaling plan has been created and applied to all
 resources.",
            "StatusCode": "Active"
        }
    ]
}
```

Pour plus d'informations, consultez What Is AWS Auto Scaling ? dans le Guide de l'utilisateur d'AWS Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous DescribeScalingPlansà la section Référence des AWS CLI commandes.

`get-scaling-plan-resource-forecast-data`

L'exemple de code suivant montre comment utiliser`get-scaling-plan-resource-forecast-data`.

AWS CLI

Pour récupérer les données de prévision de charge

Cet exemple extrait les données de prévision de charge pour une ressource évolutive (un groupe Auto Scaling) associée au plan de dimensionnement spécifié.

```
aws autoscaling-plans get-scaling-plan-resource-forecast-data \
    --scaling-plan-name my-scaling-plan \
```

```
    --scaling-plan-version 1 \
    --service-namespace "autoscaling" \
    --resource-id autoScalingGroup/my-asg \
    --scalable-dimension "autoscaling:autoScalingGroup:DesiredCapacity" \
    --forecast-data-type "LoadForecast" \
    --start-time "2019-08-30T00:00:00Z" \
    --end-time "2019-09-06T00:00:00Z"
```

Sortie :

```
{
    "Datapoints": [...]
}
```

Pour plus d'informations, consultez la section What Is AWS Auto Scaling dans le guide de l'utilisateur d'AWS Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous GetScalingPlanResourceForecastDataà la section Référence des AWS CLI commandes.

`update-scaling-plan`

L'exemple de code suivant montre comment utiliser`update-scaling-plan`.

AWS CLI

Pour mettre à jour un plan de dimensionnement

L'`update-scaling-plan`exemple suivant modifie la métrique de dimensionnement d'un groupe Auto Scaling dans le plan de dimensionnement spécifié.

```
aws autoscaling-plans update-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1 \
    --scaling-instructions
 '{"ScalableDimension":"autoscaling:autoScalingGroup:DesiredCapacity","ResourceId":"autoScal
my-asg","ServiceNamespace":"autoscaling","TargetTrackingConfigurations":
[{"PredefinedScalingMetricSpecification":
 {"PredefinedScalingMetricType":"ALBRequestCountPerTarget","ResourceLabel":"app/my-
alb/f37c06a68c1748aa/targetgroup/my-target-
group/6d4ea56ca2d6a18d"},"TargetValue":40.0}],"MinCapacity": 1,"MaxCapacity": 10}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez What Is AWS Auto Scaling ? dans le Guide de l'utilisateur d'AWS Auto Scaling.

- Pour plus de détails sur l'API, reportez-vous UpdateScalingPlanà la section Référence des AWS CLI commandes.

AWS Backup exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Backup.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-backup-plan`

L'exemple de code suivant montre comment utiliser`create-backup-plan`.

AWS CLI

Pour créer un plan de sauvegarde

L'`create-backup-plan`exemple suivant crée le plan de sauvegarde spécifié avec une rétention de 35 jours.

```
aws backup create-backup-plan \
--backup-plan "{\"BackupPlanName\":\"Example-Backup-Plan\",\"Rules\":[{\"RuleName\":
\"DailyBackups\",\"ScheduleExpression\":\"cron(0 5 ? * * *)\",\"StartWindowMinutes
```

```
\":480,\"TargetBackupVaultName\":\"Default\",\"Lifecycle\":{\"DeleteAfterDays
\":35}}]}"
```

Sortie :

```
{
    "BackupPlanId": "1fa3895c-a7f5-484a-a371-2dd6a1a9f729",
    "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:1fa3895c-
a7f5-484a-a371-2dd6a1a9f729",
    "CreationDate": 1568928754.747,
    "VersionId": "ZjQ2ZTI5YWQtZDg5Yi00MzYzLWJmZTAtMDI1MzhlMDhjYjEz"
}
```

Pour plus d'informations, consultez la section Création d'un plan de sauvegarde dans le AWS Backup Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateBackupPlanà la section Référence des AWS CLI commandes.

`create-backup-vault`

L'exemple de code suivant montre comment utiliser`create-backup-vault`.

AWS CLI

Pour créer un coffre-fort de sauvegarde

L'`create-backup-vault`exemple suivant crée un coffre-fort de sauvegarde portant le nom spécifié.

```
aws backup create-backup-vault
    --backup-vault-name sample-vault
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "BackupVaultName": "sample-vault",
    "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-
vault",
    "CreationDate": 1568928338.385
```

```
}
```

Pour plus d'informations, consultez la section Création d'un coffre-fort de sauvegarde dans le manuel AWS Backup Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateBackupVaultà la section Référence des AWS CLI commandes.

`get-backup-plan-from-template`

L'exemple de code suivant montre comment utiliser`get-backup-plan-from-template`.

AWS CLI

Pour obtenir un plan de sauvegarde existant à partir d'un modèle

L'`get-backup-plan-from-template`exemple suivant extrait un plan de sauvegarde existant à partir d'un modèle qui spécifie une sauvegarde quotidienne avec une durée de conservation de 35 jours.

```
aws backup get-backup-plan-from-template \
    --backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

Sortie :

```
{
    "BackupPlanDocument": {
        "Rules": [
            {
                "RuleName": "DailyBackups",
                "ScheduleExpression": "cron(0 5 ? * * *)",
                "StartWindowMinutes": 480,
                "Lifecycle": {
                    "DeleteAfterDays": 35
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'un plan de sauvegarde dans le AWS Backup Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetBackupPlanFromTemplateà la section Référence des AWS CLI commandes.

`get-backup-plan`

L'exemple de code suivant montre comment utiliser`get-backup-plan`.

AWS CLI

Pour obtenir les détails d'un plan de sauvegarde

L'`get-backup-plan`exemple suivant affiche les détails du plan de sauvegarde spécifié.

```
aws backup get-backup-plan \
    --backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"
```

Sortie :

```
{
    "BackupPlan": {
        "BackupPlanName": "Example-Backup-Plan",
        "Rules": [
            {
                "RuleName": "DailyBackups",
                "TargetBackupVaultName": "Default",
                "ScheduleExpression": "cron(0 5 ? * * *)",
                "StartWindowMinutes": 480,
                "CompletionWindowMinutes": 10080,
                "Lifecycle": {
                    "DeleteAfterDays": 35
                },
                "RuleId": "70e0ccdc-e9df-4e83-82ad-c1e5a9471cc3"
            }
        ]
    },
    "BackupPlanId": "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5",
    "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:fcbf5d8f-
bd77-4f3a-9c97-f24fb3d373a5",
    "VersionId": "NjQ2ZTZkODktMGVhNy00MmQ0LWE4YjktZTkwNTQ3OTkyYTcw",
    "CreationDate": 1568926091.57
}
```

Pour plus d'informations, consultez la section Création d'un plan de sauvegarde dans le AWS Backup Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetBackupPlanà la section Référence des AWS CLI commandes.

`list-backup-jobs`

L'exemple de code suivant montre comment utiliser`list-backup-jobs`.

AWS CLI

Exemple 1 : pour répertorier toutes les tâches de sauvegarde

L'`list-backup-jobs`exemple suivant renvoie des métadonnées relatives à vos tâches de sauvegarde dans votre AWS compte.

```
aws backup list-backup-jobs
```

Sortie :

```
{
    "BackupJobs": [
        {
            "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "BackupVaultName": "Default",
            "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-12345678901234567",
            "CreationDate": 1600721892.929,
            "State": "CREATED",
            "PercentDone": "0.0",
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
            "StartBy": 1600725492.929,
            "ResourceType": "EC2"
        },
        {
            "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "BackupVaultName": "Default",
            "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
```

```
            "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
            "CreationDate": 1600721724.77,
            "CompletionDate": 1600721744.488,
            "State": "COMPLETED",
            "PercentDone": "100.0",
            "BackupSizeInBytes": 71,
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
            "StartBy": 1600725324.77,
            "ResourceType": "EFS"
        }
    ]
}
```

Pour plus d'informations, consultez Creating a Backup dans le AWS Backup Developer Guide.

Exemple 2 : pour répertorier les tâches de sauvegarde terminées

L'`list-backup-jobs`exemple suivant renvoie des métadonnées relatives à vos tâches de sauvegarde terminées dans votre AWS compte.

```
aws backup list-backup-jobs \
    --by-state COMPLETED
```

Sortie :

```
{
    "BackupJobs": [
        {
            "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "BackupVaultName": "Default",
            "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
            "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
            "CreationDate": 1600721724.77,
            "CompletionDate": 1600721744.488,
            "State": "COMPLETED",
```

```
            "PercentDone": "100.0",
            "BackupSizeInBytes": 71,
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
            "StartBy": 1600725324.77,
            "ResourceType": "EFS"
        }
    ]
}
```

Pour plus d'informations, consultez Creating a Backup dans le AWS Backup Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListBackupJobsà la section Référence des AWS CLI commandes.

AWS Batch exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Batch.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-job`

L'exemple de code suivant montre comment utiliser`cancel-job`.

AWS CLI

Pour annuler une offre d'emploi

Cet exemple annule une tâche avec l'ID de tâche spécifié.

Commande :

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --
reason "Cancelling job."
```

- Pour plus de détails sur l'API, voir CancelJobla section Référence des AWS CLI commandes.

`create-compute-environment`

L'exemple de code suivant montre comment utiliser`create-compute-environment`.

AWS CLI

Pour créer un environnement informatique géré avec des instances à la demande

Cet exemple crée un environnement informatique géré avec des types d'instances C4 spécifiques lancés à la demande. L'environnement informatique s'appelle C4OnDemand.

Commande :

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
C4OnDemand.json
```

Format de fichier JSON :

```
{
  "computeEnvironmentName": "C4OnDemand",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "type": "EC2",
    "minvCpus": 0,
    "maxvCpus": 128,
    "desiredvCpus": 48,
    "instanceTypes": [
      "c4.large",
      "c4.xlarge",
      "c4.2xlarge",
      "c4.4xlarge",
      "c4.8xlarge"
    ],
```

```
    "subnets": [
      "subnet-220c0e0a",
      "subnet-1a95556d",
      "subnet-978f6dce"
    ],
    "securityGroupIds": [
      "sg-cf5093b2"
    ],
    "ec2KeyPair": "id_rsa",
    "instanceRole": "ecsInstanceRole",
    "tags": {
      "Name": "Batch Instance - C4OnDemand"
    }
  },
  "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

Sortie :

```
{
    "computeEnvironmentName": "C4OnDemand",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

Pour créer un environnement informatique géré avec des instances Spot

Cet exemple crée un environnement informatique géré avec le type d'instance M4 qui est lancé lorsque le prix de l'offre au comptant est égal ou inférieur à 20 % du prix à la demande pour le type d'instance. L'environnement informatique s'appelle M4Spot.

Commande :

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
M4Spot.json
```

Format de fichier JSON :

```
{
  "computeEnvironmentName": "M4Spot",
  "type": "MANAGED",
```

```
  "state": "ENABLED",
  "computeResources": {
    "type": "SPOT",
    "spotIamFleetRole": "arn:aws:iam::012345678910:role/aws-ec2-spot-fleet-role",
    "minvCpus": 0,
    "maxvCpus": 128,
    "desiredvCpus": 4,
    "instanceTypes": [
      "m4"
    ],
    "bidPercentage": 20,
    "subnets": [
      "subnet-220c0e0a",
      "subnet-1a95556d",
      "subnet-978f6dce"
    ],
    "securityGroupIds": [
      "sg-cf5093b2"
    ],
    "ec2KeyPair": "id_rsa",
    "instanceRole": "ecsInstanceRole",
    "tags": {
      "Name": "Batch Instance - M4Spot"
    }
  },
  "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

Sortie :

```
{
    "computeEnvironmentName": "M4Spot",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/M4Spot"
}
```

- Pour plus de détails sur l'API, voir CreateComputeEnvironmentla section Référence des AWS CLI commandes.

`create-job-queue`

L'exemple de code suivant montre comment utiliser`create-job-queue`.

AWS CLI

Pour créer une file d'attente de tâches à faible priorité avec un environnement informatique unique

Cet exemple crée une file d'attente de tâches appelée LowPriority qui utilise l'environnement informatique M4Spot.

Commande :

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
LowPriority.json
```

Format de fichier JSON :

```
{
  "jobQueueName": "LowPriority",
  "state": "ENABLED",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

Sortie :

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",
    "jobQueueName": "LowPriority"
}
```

Pour créer une file d'attente de tâches hautement prioritaires avec deux environnements informatiques

Cet exemple crée une file d'attente de tâches appelée HighPriority qui utilise l'environnement OnDemand informatique C4 avec un ordre de 1 et l'environnement de calcul M4Spot avec un ordre de 2. Le planificateur essaiera d'abord de placer les tâches dans l'environnement OnDemand informatique C4.

Commande :

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
HighPriority.json
```

Format de fichier JSON :

```
{
  "jobQueueName": "HighPriority",
  "state": "ENABLED",
  "priority": 1,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "C4OnDemand"
    },
    {
      "order": 2,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

Sortie :

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
    "jobQueueName": "HighPriority"
}
```

- Pour plus de détails sur l'API, voir CreateJobQueuela section Référence des AWS CLI commandes.

`delete-compute-environment`

L'exemple de code suivant montre comment utiliser`delete-compute-environment`.

AWS CLI

Pour supprimer un environnement informatique

Cet exemple supprime l'environnement de OnDemand calcul P2.

Commande :

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- Pour plus de détails sur l'API, voir DeleteComputeEnvironmentla section Référence des AWS CLI commandes.

`delete-job-queue`

L'exemple de code suivant montre comment utiliser`delete-job-queue`.

AWS CLI

Pour supprimer une file d'attente de tâches

Cet exemple supprime la file d'attente des tâches GPGPU.

Commande :

```
aws batch delete-job-queue --job-queue GPGPU
```

- Pour plus de détails sur l'API, reportez-vous DeleteJobQueueà la section Référence des AWS CLI commandes.

`deregister-job-definition`

L'exemple de code suivant montre comment utiliser`deregister-job-definition`.

AWS CLI

Pour annuler l'enregistrement d'une définition de tâche

Cet exemple annule l'enregistrement d'une définition de tâche appelée sleep10.

Commande :

```
aws batch deregister-job-definition --job-definition sleep10
```

- Pour plus de détails sur l'API, reportez-vous DeregisterJobDefinitionà la section Référence des AWS CLI commandes.

`describe-compute-environments`

L'exemple de code suivant montre comment utiliser`describe-compute-environments`.

AWS CLI

Pour décrire un environnement informatique

Cet exemple décrit l'environnement de OnDemand calcul P2.

Commande :

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

Sortie :

```
{
    "computeEnvironments": [
        {
            "status": "VALID",
            "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole",
            "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand",
            "computeResources": {
                "subnets": [
                    "subnet-220c0e0a",
                    "subnet-1a95556d",
                    "subnet-978f6dce"
                ],
                "tags": {
                    "Name": "Batch Instance - P2OnDemand"
                },
                "desiredvCpus": 48,
                "minvCpus": 0,
                "instanceTypes": [
                    "p2"
                ],
                "securityGroupIds": [
                    "sg-cf5093b2"
```

```
                ],
                "instanceRole": "ecsInstanceRole",
                "maxvCpus": 128,
                "type": "EC2",
                "ec2KeyPair": "id_rsa"
            },
            "statusReason": "ComputeEnvironment Healthy",
            "ecsClusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/
P2OnDemand_Batch_2c06f29d-d1fe-3a49-879d-42394c86effc",
            "state": "ENABLED",
            "computeEnvironmentName": "P2OnDemand",
            "type": "MANAGED"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeComputeEnvironmentsà la section Référence des AWS CLI commandes.

`describe-job-definitions`

L'exemple de code suivant montre comment utiliser`describe-job-definitions`.

AWS CLI

Pour décrire les définitions de tâches actives

Cet exemple décrit toutes vos définitions de tâches actives.

Commande :

```
aws batch describe-job-definitions --status ACTIVE
```

Sortie :

```
{
    "jobDefinitions": [
        {
            "status": "ACTIVE",
            "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-
definition/sleep60:1",
            "containerProperties": {
```

```
                "mountPoints": [],
                "parameters": {},
                "image": "busybox",
                "environment": {},
                "vcpus": 1,
                "command": [
                    "sleep",
                    "60"
                ],
                "volumes": [],
                "memory": 128,
                "ulimits": []
            },
            "type": "container",
            "jobDefinitionName": "sleep60",
            "revision": 1
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeJobDefinitionsà la section Référence des AWS CLI commandes.

`describe-job-queues`

L'exemple de code suivant montre comment utiliser`describe-job-queues`.

AWS CLI

Pour décrire une file d'attente de tâches

Cet exemple décrit la file HighPriority d'attente des tâches.

Commande :

```
aws batch describe-job-queues --job-queues HighPriority
```

Sortie :

```
{
    "jobQueues": [
        {
```

```
            "status": "VALID",
            "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/
HighPriority",
            "computeEnvironmentOrder": [
                {
                    "computeEnvironment": "arn:aws:batch:us-
east-1:012345678910:compute-environment/C4OnDemand",
                    "order": 1
                }
            ],
            "statusReason": "JobQueue Healthy",
            "priority": 1,
            "state": "ENABLED",
            "jobQueueName": "HighPriority"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeJobQueuesà la section Référence des AWS CLI commandes.

`describe-jobs`

L'exemple de code suivant montre comment utiliser`describe-jobs`.

AWS CLI

Pour décrire un poste

L'`describe-jobs`exemple suivant décrit une tâche avec l'ID de tâche spécifié.

```
aws batch describe-jobs \
    --jobs bcf0b186-a532-4122-842e-2ccab8d54efb
```

Sortie :

```
{
    "jobs": [
        {
            "status": "SUBMITTED",
            "container": {
                "mountPoints": [],
```

```
                "image": "busybox",
                "environment": [],
                "vcpus": 1,
                "command": [
                    "sleep",
                    "60"
                ],
                "volumes": [],
                "memory": 128,
                "ulimits": []
            },
            "parameters": {},
            "jobDefinition": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep60:1",
            "jobQueue": "arn:aws:batch:us-east-1:012345678910:job-queue/
HighPriority",
            "jobId": "bcf0b186-a532-4122-842e-2ccab8d54efb",
            "dependsOn": [],
            "jobName": "example",
            "createdAt": 1480483387803
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeJobsà la section Référence des AWS CLI commandes.

`list-jobs`

L'exemple de code suivant montre comment utiliser`list-jobs`.

AWS CLI

Pour répertorier les tâches en cours

Cet exemple répertorie les tâches en cours dans la file HighPriority d'attente des tâches.

Commande :

```
aws batch list-jobs --job-queue HighPriority
```

Sortie :

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
        }
    ]
}
```

Pour répertorier les offres d'emploi soumises

Cet exemple répertorie les HighPriority tâches de la file d'attente dont le statut est SOUMIS.

Commande :

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

Sortie :

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListJobsà la section Référence des AWS CLI commandes.

`register-job-definition`

L'exemple de code suivant montre comment utiliser`register-job-definition`.

AWS CLI

Pour enregistrer une définition de tâche

Cet exemple enregistre une définition de tâche pour une tâche de conteneur simple.

Commande :

```
aws batch register-job-definition --job-definition-name sleep30 --type container --
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":
 [ "sleep", "30"]}'
```

Sortie :

```
{
    "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep30:1",
    "jobDefinitionName": "sleep30",
    "revision": 1
}
```

- Pour plus de détails sur l'API, reportez-vous RegisterJobDefinitionà la section Référence des AWS CLI commandes.

`submit-job`

L'exemple de code suivant montre comment utiliser`submit-job`.

AWS CLI

Pour soumettre une offre d'emploi

Cet exemple soumet une tâche de conteneur simple appelée example à la file d'attente des HighPriority tâches.

Commande :

```
aws batch submit-job --job-name example --job-queue HighPriority  --job-
definition sleep60
```

Sortie :

```
{
    "jobName": "example",
    "jobId": "876da822-4198-45f2-a252-6cea32512ea8"
}
```

- Pour plus de détails sur l'API, reportez-vous SubmitJobà la section Référence des AWS CLI commandes.

`terminate-job`

L'exemple de code suivant montre comment utiliser`terminate-job`.

AWS CLI

Pour mettre fin à une tâche

Cet exemple met fin à une tâche avec l'ID de tâche spécifié.

Commande :

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --
reason "Terminating job."
```

- Pour plus de détails sur l'API, reportez-vous TerminateJobà la section Référence des AWS CLI commandes.

`update-compute-environment`

L'exemple de code suivant montre comment utiliser`update-compute-environment`.

AWS CLI

Pour mettre à jour un environnement informatique

Cet exemple désactive l'environnement OnDemand informatique P2 afin qu'il puisse être supprimé.

Commande :

```
aws batch update-compute-environment --compute-environment P2OnDemand --
state DISABLED
```

Sortie :

```
{
    "computeEnvironmentName": "P2OnDemand",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateComputeEnvironmentà la section Référence des AWS CLI commandes.

update-job-queue

L'exemple de code suivant montre comment utiliserupdate-job-queue.

AWS CLI

Pour mettre à jour une file d'attente de tâches

Cet exemple désactive une file de tâches afin de pouvoir la supprimer.

Commande :

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

Sortie :

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",
    "jobQueueName": "GPGPU"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateJobQueueà la section Référence des AWS CLI commandes.

AWS Budgets exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Budgets.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-budget`

L'exemple de code suivant montre comment utiliser`create-budget`.

AWS CLI

Pour créer un budget de coûts et d'utilisation

La `create-budget` commande suivante crée un budget de coût et d'utilisation.

```
aws budgets create-budget \
    --account-id 111122223333 \
    --budget file://budget.json \
    --notifications-with-subscribers file://notifications-with-subscribers.json
```

Contenu de `budget.json` :

```
{
    "BudgetLimit": {
        "Amount": "100",
        "Unit": "USD"
    },
    "BudgetName": "Example Tag Budget",
    "BudgetType": "COST",
    "CostFilters": {
        "TagKeyValue": [
            "user:Key$value1",
            "user:Key$value2"
        ]
    },
    "CostTypes": {
        "IncludeCredit": true,
        "IncludeDiscount": true,
        "IncludeOtherSubscription": true,
        "IncludeRecurring": true,
        "IncludeRefund": true,
        "IncludeSubscription": true,
        "IncludeSupport": true,
        "IncludeTax": true,
```

```
        "IncludeUpfront": true,
        "UseBlended": false
    },
    "TimePeriod": {
        "Start": 1477958399,
        "End": 3706473600
    },
    "TimeUnit": "MONTHLY"
}
```

Contenu de `notifications-with-subscribers.json` :

```
[
    {
        "Notification": {
            "ComparisonOperator": "GREATER_THAN",
            "NotificationType": "ACTUAL",
            "Threshold": 80,
            "ThresholdType": "PERCENTAGE"
        },
        "Subscribers": [
            {
                "Address": "example@example.com",
                "SubscriptionType": "EMAIL"
            }
        ]
    }
]
```

- Pour plus de détails sur l'API, reportez-vous CreateBudgetà la section Référence des AWS CLI commandes.

`create-notification`

L'exemple de code suivant montre comment utiliser`create-notification`.

AWS CLI

Pour créer une notification pour le budget de coût et d'utilisation spécifié

Cet exemple crée une notification pour le budget de coût et d'utilisation spécifié.

Commande :

```
aws budgets create-notification --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- Pour plus de détails sur l'API, reportez-vous CreateNotificationà la section Référence des AWS CLI commandes.

`create-subscriber`

L'exemple de code suivant montre comment utiliser`create-subscriber`.

AWS CLI

Pour créer un abonné pour une notification associée à un budget de coûts et d'utilisation

Cet exemple crée un abonné pour la notification spécifiée.

Commande :

```
aws budgets create-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- Pour plus de détails sur l'API, reportez-vous CreateSubscriberà la section Référence des AWS CLI commandes.

`delete-budget`

L'exemple de code suivant montre comment utiliser`delete-budget`.

AWS CLI

Pour supprimer un budget de coûts et d'utilisation

Cet exemple supprime le budget de coût et d'utilisation spécifié.

Commande :

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- Pour plus de détails sur l'API, reportez-vous DeleteBudgetà la section Référence des AWS CLI commandes.

`delete-notification`

L'exemple de code suivant montre comment utiliser`delete-notification`.

AWS CLI

Pour supprimer une notification d'un budget

Cet exemple supprime la notification spécifiée du budget spécifié.

Commande :

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
```

- Pour plus de détails sur l'API, reportez-vous DeleteNotificationà la section Référence des AWS CLI commandes.

`delete-subscriber`

L'exemple de code suivant montre comment utiliser`delete-subscriber`.

AWS CLI

Pour supprimer un abonné d'une notification

Cet exemple supprime l'abonné spécifié de la notification spécifiée.

Commande :

```
aws budgets delete-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriberà la section Référence des AWS CLI commandes.

`describe-budget`

L'exemple de code suivant montre comment utiliser`describe-budget`.

AWS CLI

Pour récupérer un budget associé à un compte

Cet exemple extrait le budget de coût et d'utilisation spécifié.

Commande :

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

Sortie :


```
{
   "Budget": {
       "CalculatedSpend": {
           "ForecastedSpend": {
               "Amount": "2641.548000000000229192664846777915954589843 75",
               "Unit": "USD"
           },
           "ActualSpend": {
               "Amount": "604.456000000000017280399333685636520385742 1875",
               "Unit": "USD"
           }
       },
       "BudgetType": "COST",
       "BudgetLimit": {
           "Amount": "100",
           "Unit": "USD"
       },
       "BudgetName": "Example Budget",
       "CostTypes": {
           "IncludeOtherSubscription": true,
           "IncludeUpfront": true,
           "IncludeRefund": true,
           "UseBlended": false,
           "IncludeDiscount": true,
           "UseAmortized": false,
           "IncludeTax": true,
           "IncludeCredit": true,
```

```
            "IncludeSupport": true,
            "IncludeRecurring": true,
            "IncludeSubscription": true
        },
        "TimeUnit": "MONTHLY",
        "TimePeriod": {
            "Start": 1477958399.0,
            "End": 3706473600.0
        },
        "CostFilters": {
            "AZ": [
                "us-east-1"
            ]
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeBudgetà la section Référence des AWS CLI commandes.

`describe-budgets`

L'exemple de code suivant montre comment utiliser`describe-budgets`.

AWS CLI

Pour récupérer les budgets associés à un compte

Cet exemple permet de récupérer les budgets de coût et d'utilisation d'un compte.

Commande :

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

Sortie :

```
{
    "Budgets": [
        {
            "CalculatedSpend": {
                "ForecastedSpend": {
```

```
                    "Amount": "2641.548000000000229192664846777915954589843 75",
                    "Unit": "USD"
                },
                "ActualSpend": {
                    "Amount": "604.456000000000017280399333685636520385742 1875",
                    "Unit": "USD"
                }
            },
            "BudgetType": "COST",
            "BudgetLimit": {
                "Amount": "100",
                "Unit": "USD"
            },
            "BudgetName": "Example Budget",
            "CostTypes": {
                "IncludeOtherSubscription": true,
                "IncludeUpfront": true,
                "IncludeRefund": true,
                "UseBlended": false,
                "IncludeDiscount": true,
                "UseAmortized": false,
                "IncludeTax": true,
                "IncludeCredit": true,
                "IncludeSupport": true,
                "IncludeRecurring": true,
                "IncludeSubscription": true
            },
            "TimeUnit": "MONTHLY",
            "TimePeriod": {
                "Start": 1477958399.0,
                "End": 3706473600.0
            },
            "CostFilters": {
                "AZ": [
                    "us-east-1"
                ]
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeBudgetsà la section Référence des AWS CLI commandes.

`describe-notifications-for-budget`

L'exemple de code suivant montre comment utiliser`describe-notifications-for-budget`.

AWS CLI

Pour récupérer les notifications relatives à un budget

Cet exemple permet de récupérer les notifications relatives à un budget de coût et d'utilisation.

Commande :

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-name "Example Budget" --max-results 5
```

Sortie :

```
{
   "Notifications": [
       {
           "Threshold": 80.0,
           "ComparisonOperator": "GREATER_THAN",
           "NotificationType": "ACTUAL"
       }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeNotificationsForBudgetà la section Référence des AWS CLI commandes.

`describe-subscribers-for-notification`

L'exemple de code suivant montre comment utiliser`describe-subscribers-for-notification`.

AWS CLI

Pour récupérer les abonnés pour une notification de budget

Cet exemple récupère les abonnés pour une notification de budget de coût et d'utilisation.

Commande :

```
aws budgets describe-subscribers-for-notification --
account-id 111122223333 --budget-name "Example Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --max-results 5
```

Sortie :

```
{
   "Subscribers": [
       {
           "SubscriptionType": "EMAIL",
           "Address": "example2@example.com"
       },
       {
           "SubscriptionType": "EMAIL",
           "Address": "example@example.com"
       }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSubscribersForNotificationà la section Référence des AWS CLI commandes.

`update-budget`

L'exemple de code suivant montre comment utiliser`update-budget`.

AWS CLI

Pour remplacer un budget par un budget de coûts et d'utilisation

Cet exemple remplace un budget de coûts et d'utilisation par un nouveau budget.

Commande :

```
aws budgets update-budget --account-id 111122223333 --new-budget file://new-
budget.json
```

new-budget.json :

```
{
    "BudgetLimit": {
```

```
        "Amount": "100",
        "Unit": "USD"
    },
    "BudgetName": "Example Budget",
    "BudgetType": "COST",
    "CostFilters": {
        "AZ" : [ "us-east-1" ]
    },
    "CostTypes": {
        "IncludeCredit": false,
        "IncludeDiscount": true,
        "IncludeOtherSubscription": true,
        "IncludeRecurring": true,
        "IncludeRefund": true,
        "IncludeSubscription": true,
        "IncludeSupport": true,
        "IncludeTax": true,
        "IncludeUpfront": true,
        "UseBlended": false,
        "UseAmortized": true
    },
    "TimePeriod": {
        "Start": 1477958399,
        "End": 3706473600
    },
    "TimeUnit": "MONTHLY"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateBudgetà la section Référence des AWS CLI commandes.

`update-notification`

L'exemple de code suivant montre comment utiliser`update-notification`.

AWS CLI

Pour remplacer une notification relative à un budget de coûts et d'utilisation

Cet exemple remplace une notification de 80 % pour un budget de coûts et d'utilisation par une notification de 90 %.

Commande :

```
aws budgets update-notification --account-id 111122223333 --budget-name "Example
 Budget" --old-notification
  NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT
 --new-notification
  NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=90,ThresholdType=PERCENT
```

- Pour plus de détails sur l'API, reportez-vous UpdateNotificationà la section Référence des AWS CLI commandes.

`update-subscriber`

L'exemple de code suivant montre comment utiliser`update-subscriber`.

AWS CLI

Pour remplacer un abonné pour un budget de coûts et d'utilisation

Cet exemple remplace l'abonné pour un budget de coût et d'utilisation.

Commande :

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-
subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- Pour plus de détails sur l'API, reportez-vous UpdateSubscriberà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Chime AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Chime.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-phone-number-with-user`

L'exemple de code suivant montre comment utiliser`associate-phone-number-with-user`.

AWS CLI

Pour associer un numéro de téléphone à un utilisateur

L'`associate-phone-number-with-user`exemple suivant associe le numéro de téléphone spécifié à un utilisateur.

```
aws chime associate-phone-number-with-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \
    --e164-phone-number "+12065550100"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des numéros de téléphone des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous AssociatePhoneNumberWithUserà la section Référence des AWS CLI commandes.

`associate-signin-delegate-groups-with-account`

L'exemple de code suivant montre comment utiliser`associate-signin-delegate-groups-with-account`.

AWS CLI

Pour associer des groupes de délégués à la connexion

L'`associate-signin-delegate-groups-with-account`exemple suivant associe le groupe de délégués de connexion spécifié au compte Amazon Chime spécifié.

```
aws chime associate-signin-delegate-groups-with-account \
```

```
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --signin-delegate-groups GroupName=my_users
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de l'accès et des autorisations des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous AssociateSigninDelegateGroupsWithAccountà la section Référence des AWS CLI commandes.

`batch-create-room-membership`

L'exemple de code suivant montre comment utiliser`batch-create-room-membership`.

AWS CLI

Pour créer plusieurs abonnements à des salles

L'`batch-create-room-membership`exemple suivant ajoute plusieurs utilisateurs à un salon de discussion en tant que membres du salon de discussion. Il attribue également des rôles d'administrateur et de membre aux utilisateurs.

```
aws chime batch-create-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --membership-item-list "MemberId=1ab2345c-67de-8901-
f23g-45h678901j2k,Role=Administrator" "MemberId=2ab2345c-67de-8901-
f23g-45h678901j2k,Role=Member"
```

Sortie :

```
{
    "ResponseMetadata": {
        "RequestId": "169ba401-d886-475f-8b3f-e01eac6fadfb",
        "HTTPStatusCode": 201,
        "HTTPHeaders": {
            "x-amzn-requestid": "169ba401-d886-475f-8b3f-e01eac6fadfb",
            "content-type": "application/json",
            "content-length": "13",
            "date": "Mon, 02 Dec 2019 22:46:58 GMT",
            "connection": "keep-alive"
```

```
        },
        "RetryAttempts": 0
    },
    "Errors": []
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous BatchCreateRoomMembershipà la section Référence des AWS CLI commandes.

`batch-delete-phone-number`

L'exemple de code suivant montre comment utiliser`batch-delete-phone-number`.

AWS CLI

Pour supprimer plusieurs numéros de téléphone

L'`batch-delete-phone-number`exemple suivant supprime tous les numéros de téléphone spécifiés.

```
aws chime batch-delete-phone-number \
    --phone-number-ids "%2B12065550100" "%2B12065550101"
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "PhoneNumberErrors": []
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous BatchDeletePhoneNumberà la section Référence des AWS CLI commandes.

`batch-suspend-user`

L'exemple de code suivant montre comment utiliser`batch-suspend-user`.

AWS CLI

Pour suspendre plusieurs utilisateurs

L'`batch-suspend-user`exemple suivant suspend les utilisateurs listés du compte Amazon Chime spécifié.

```
aws chime batch-suspend-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

Sortie :

```
{
    "UserErrors": []
}
```

- Pour plus de détails sur l'API, reportez-vous BatchSuspendUserà la section Référence des AWS CLI commandes.

`batch-unsuspend-user`

L'exemple de code suivant montre comment utiliser`batch-unsuspend-user`.

AWS CLI

Pour annuler la suspension de plusieurs utilisateurs

L'`batch-unsuspend-user`exemple suivant supprime toute suspension antérieure pour les utilisateurs répertoriés sur le compte Amazon Chime spécifié.

```
aws chime batch-unsuspend-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

Sortie :

```
{
    "UserErrors": []
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous BatchUnsuspendUserà la section Référence des AWS CLI commandes.

`batch-update-phone-number`

L'exemple de code suivant montre comment utiliser`batch-update-phone-number`.

AWS CLI

Pour mettre à jour plusieurs numéros de téléphone, types de produits en même temps

L'`batch-update-phone-number`exemple suivant met à jour les types de produits pour tous les numéros de téléphone spécifiés.

```
aws chime batch-update-phone-number \
    --update-phone-number-request-items PhoneNumberId=
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=
%2B12065550101,ProductType=BusinessCalling
```

Sortie :

```
{
    "PhoneNumberErrors": []
}
```

Pour mettre à jour les noms d'appel de plusieurs numéros de téléphone en même temps

L'`batch-update-phone-number`exemple suivant met à jour les noms d'appel de tous les numéros de téléphone spécifiés.

```
aws chime batch-update-phone-number \
    --update-phone-number-request-items PhoneNumberId=
%2B14013143874,CallingName=phonenumber1 PhoneNumberId=
%2B14013144061,CallingName=phonenumber2
```

Sortie :

```
{
    "PhoneNumberErrors": []
```

```
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous BatchUpdatePhoneNumberà la section Référence des AWS CLI commandes.

`batch-update-user`

L'exemple de code suivant montre comment utiliser`batch-update-user`.

AWS CLI

Pour mettre à jour plusieurs utilisateurs en une seule commande

L'`batch-update-user`exemple suivant met à jour le `LicenseType` pour chacun des utilisateurs répertoriés dans le compte Amazon Chime spécifié.

```
aws chime batch-update-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
    --update-user-request-items "UserId=a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE,LicenseType=Basic" "UserId=a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE,LicenseType=Basic"
```

Sortie :

```
{
    "UserErrors": []
}
```

- Pour plus de détails sur l'API, reportez-vous BatchUpdateUserà la section Référence des AWS CLI commandes.

`create-account`

L'exemple de code suivant montre comment utiliser`create-account`.

AWS CLI

Pour créer un compte

L'`create-account`exemple suivant crée un compte Amazon Chime sous le compte de l' AWS administrateur.

```
aws chime create-account \
    --name MyChimeAccount
```

Sortie :

```
{
    "Account": {
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "MyChimeAccount",
        "AccountType": "Team",
        "CreatedTimestamp": "2019-01-04T17:11:22.003Z",
        "DefaultLicense": "Pro",
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
                "GroupName": "myGroup"
            },
        ]
    }
}
```

Pour plus d'informations, consultez Getting Started dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous CreateAccountà la section Référence des AWS CLI commandes.

`create-bot`

L'exemple de code suivant montre comment utiliser`create-bot`.

AWS CLI

Pour créer un bot Amazon Chime

L'`create-bot`exemple suivant crée un bot pour le compte Amazon Chime Enterprise spécifié.

```
aws chime create-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --display-name "myBot" \
    --domain "example.com"
```

Sortie :

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
}
```

Pour plus d'informations, consultez Intégrer un chatbot à Amazon Chime dans le manuel Amazon Chime Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateBotà la section Référence des AWS CLI commandes.

`create-phone-number-order`

L'exemple de code suivant montre comment utiliser`create-phone-number-order`.

AWS CLI

Pour créer une commande par numéro de téléphone

L'`create-phone-number-order`exemple suivant crée un ordre de numéros de téléphone pour les numéros de téléphone spécifiés.

```
aws chime create-phone-number-order \
```

```
    --product-type VoiceConnector \
    --e164-phone-numbers "+12065550100" "+12065550101" "+12065550102"
```

Sortie :

```
{
    "PhoneNumberOrder": {
        "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
        "ProductType": "VoiceConnector",
        "Status": "Processing",
        "OrderedPhoneNumbers": [
            {
                "E164PhoneNumber": "+12065550100",
                "Status": "Processing"
            },
            {
               "E164PhoneNumber": "+12065550101",
               "Status": "Processing"
            },
            {
              "E164PhoneNumber": "+12065550102",
              "Status": "Processing"
            }
        ],
        "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
        "UpdatedTimestamp": "2019-08-09T21:35:22.408Z"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous CreatePhoneNumberOrderà la section Référence des AWS CLI commandes.

`create-room-membership`

L'exemple de code suivant montre comment utiliser`create-room-membership`.

AWS CLI

Pour créer un abonnement à une chambre

L'`create-room-membership`exemple suivant ajoute l'utilisateur spécifié au salon de discussion en tant que membre du salon de discussion.

```
aws chime create-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Sortie :

```
{
    "RoomMembership": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Member": {
            "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
            "MemberType": "User",
            "Email": "janed@example.com",
            "FullName": "Jane Doe",
            "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
        },
        "Role": "Member",
        "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
        "UpdatedTimestamp": "2019-12-02T22:36:41.969Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous CreateRoomMembershipà la section Référence des AWS CLI commandes.

`create-room`

L'exemple de code suivant montre comment utiliser`create-room`.

AWS CLI

Pour créer un salon de discussion

L'`create-room`exemple suivant crée un salon de discussion pour le compte Amazon Chime spécifié.

```
aws chime create-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --name chatRoom
```

Sortie :

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "chatRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous CreateRoomà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un profil utilisateur pour un appareil partagé

L'`create-user`exemple suivant crée un profil d'appareil partagé pour l'adresse e-mail spécifiée.

```
aws chime create-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --email roomdevice@example.com \
    --user-type SharedDevice
```

Sortie :

```
{
```

```
    "User": {
        "UserId": "1ab2345c-67de-8901-f23g-45h678901j2k",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "PrimaryEmail": "roomdevice@example.com",
        "DisplayName": "Room Device",
        "LicenseType": "Pro",
        "UserType": "SharedDevice",
        "UserRegistrationStatus": "Registered",
        "RegisteredOn": "2020-01-15T22:38:09.806Z",
        "AlexaForBusinessMetadata": {
            "IsAlexaForBusinessEnabled": false
        }
    }
}
```

Pour plus d'informations, consultez la section Préparation à la configuration dans le Guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`delete-account`

L'exemple de code suivant montre comment utiliser`delete-account`.

AWS CLI

Pour supprimer un compte

L'`delete-account`exemple suivant supprime le compte spécifié.

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer votre compte dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous DeleteAccountà la section Référence des AWS CLI commandes.

`delete-phone-number`

L'exemple de code suivant montre comment utiliser`delete-phone-number`.

AWS CLI

Pour supprimer un numéro de téléphone

L'`delete-phone-number`exemple suivant place le numéro de téléphone spécifié dans la file d'attente de suppression.

```
aws chime delete-phone-number \
    --phone-number-id "+12065550100"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous DeletePhoneNumberà la section Référence des AWS CLI commandes.

`delete-room-membership`

L'exemple de code suivant montre comment utiliser`delete-room-membership`.

AWS CLI

Pour supprimer un utilisateur en tant que membre d'un salon de discussion

L'`delete-room-membership`exemple suivant supprime le membre spécifié de la salle de discussion spécifiée.

```
aws chime delete-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous DeleteRoomMembershipà la section Référence des AWS CLI commandes.

`delete-room`

L'exemple de code suivant montre comment utiliser`delete-room`.

AWS CLI

Pour supprimer un salon de discussion

L'`delete-room`exemple suivant supprime le salon de discussion spécifié et supprime les adhésions au salon de discussion.

```
aws chime delete-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous DeleteRoomà la section Référence des AWS CLI commandes.

`disassociate-phone-number-from-user`

L'exemple de code suivant montre comment utiliser`disassociate-phone-number-from-user`.

AWS CLI

Pour dissocier un numéro de téléphone d'un utilisateur

L'`disassociate-phone-number-from-user`exemple suivant dissocie un numéro de téléphone de l'utilisateur spécifié.

```
aws chime disassociate-phone-number-from-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des numéros de téléphone des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous DisassociatePhoneNumberFromUserà la section Référence des AWS CLI commandes.

`disassociate-signin-delegate-groups-from-account`

L'exemple de code suivant montre comment utiliser`disassociate-signin-delegate-groups-from-account`.

AWS CLI

Pour dissocier les groupes de délégués de connexion

L'`disassociate-signin-delegate-groups-from-account`exemple suivant dissocie le groupe de délégués de connexion spécifié du compte Amazon Chime spécifié.

```
aws chime disassociate-signin-delegate-groups-from-account \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --group-names "my_users"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de l'accès et des autorisations des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous DisassociateSigninDelegateGroupsFromAccountà la section Référence des AWS CLI commandes.

`get-account-settings`

L'exemple de code suivant montre comment utiliser`get-account-settings`.

AWS CLI

Pour récupérer les paramètres d'un compte

L'`get-account-settings`exemple suivant récupère les paramètres du compte spécifié.

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
    "AccountSettings": {
        "DisableRemoteControl": false,
        "EnableDialOut": false
    }
}
```

Pour plus d'informations, consultez la section Gestion de vos comptes Amazon Chime dans le Guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetAccountSettingsà la section Référence des AWS CLI commandes.

`get-account`

L'exemple de code suivant montre comment utiliser`get-account`.

AWS CLI

Pour récupérer les informations d'un compte

L'`get-account`exemple suivant récupère les détails du compte Amazon Chime spécifié.

```
aws chime get-account \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
    "Account": {
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "EnterpriseDirectory",
        "AccountType": "EnterpriseDirectory",
        "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
        "DefaultLicense": "Pro",
```

```
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
                "GroupName": "myGroup"
            },
        ]
    }
}
```

Pour plus d'informations, consultez la section Gestion de vos comptes Amazon Chime dans le Guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetAccountà la section Référence des AWS CLI commandes.

`get-bot`

L'exemple de code suivant montre comment utiliser`get-bot`.

AWS CLI

Pour récupérer les informations relatives à un bot

L'`get-bot`exemple suivant affiche les détails du bot spécifié.

```
aws chime get-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Sortie :

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
```

```
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
}
```

Pour plus d'informations, consultez la section Mettre à jour les robots de discussion dans le manuel Amazon Chime Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetBotà la section Référence des AWS CLI commandes.

`get-global-settings`

L'exemple de code suivant montre comment utiliser`get-global-settings`.

AWS CLI

Pour obtenir les paramètres globaux

L'`get-global-settings`exemple suivant extrait les noms des compartiments S3 utilisés pour stocker les enregistrements détaillés des appels pour Amazon Chime Business Calling et Amazon Chime Voice Connectors associés au compte de l'administrateur. AWS

```
aws chime get-global-settings
```

Sortie :

```
{
    "BusinessCalling": {
        "CdrBucket": "s3bucket"
    },
    "VoiceConnector": {
        "CdrBucket": "s3bucket"
    }
}
```

Pour plus d'informations, consultez la section Gestion des paramètres globaux dans le Guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetGlobalSettingsà la section Référence des AWS CLI commandes.

`get-phone-number-order`

L'exemple de code suivant montre comment utiliser`get-phone-number-order`.

AWS CLI

Pour obtenir les détails d'une commande par numéro de téléphone

L'`get-phone-number-order`exemple suivant affiche les détails de la commande de numéros de téléphone spécifiée.

```
aws chime get-phone-number-order \
    --phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

Sortie :

```
{
    "PhoneNumberOrder": {
        "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
        "ProductType": "VoiceConnector",
        "Status": "Partial",
        "OrderedPhoneNumbers": [
            {
              "E164PhoneNumber": "+12065550100",
              "Status": "Acquired"
            },
            {
                "E164PhoneNumber": "+12065550101",
                "Status": "Acquired"
            },
            {
                "E164PhoneNumber": "+12065550102",
                "Status": "Failed"
            }
        ],
        "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
        "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetPhoneNumberOrderà la section Référence des AWS CLI commandes.

`get-phone-number-settings`

L'exemple de code suivant montre comment utiliser`get-phone-number-settings`.

AWS CLI

Pour récupérer le nom d'un appel sortant

L'`get-phone-number-settings`exemple suivant récupère le nom d'appel sortant par défaut pour le compte de l'utilisateur appelant AWS .

```
aws chime get-phone-number-settings
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "CallingName": "myName",
    "CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetPhoneNumberSettingsà la section Référence des AWS CLI commandes.

`get-phone-number`

L'exemple de code suivant montre comment utiliser`get-phone-number`.

AWS CLI

Pour obtenir les détails du numéro de téléphone

L'`get-phone-number`exemple suivant affiche les détails du numéro de téléphone spécifié.

```
aws chime get-phone-number \
    --phone-number-id +12065550100
```

Sortie :

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "VoiceConnector",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
            "OutboundMMS": true
        },
       "Associations": [
            {
                "Value": "abcdef1ghij2klmno3pqr4",
                "Name": "VoiceConnectorId",
                "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
            }
        ],
        "CallingNameStatus": "UpdateInProgress",
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-09T21:35:31.745Z"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetPhoneNumberà la section Référence des AWS CLI commandes.

`get-room`

L'exemple de code suivant montre comment utiliser`get-room`.

AWS CLI

Pour obtenir des informations sur un salon de discussion

L'`get-room`exemple suivant affiche des détails sur le salon de discussion spécifié.

```
aws chime get-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Sortie :

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "chatRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetRoomà la section Référence des AWS CLI commandes.

`get-user-settings`

L'exemple de code suivant montre comment utiliser`get-user-settings`.

AWS CLI

Pour récupérer les paramètres utilisateur

L'`get-user-settings`exemple suivant affiche les paramètres utilisateur spécifiés.

```
aws chime get-user-settings \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
```

```
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Sortie :

```
{
    "UserSettings": {
        "Telephony": {
            "InboundCalling": true,
            "OutboundCalling": true,
            "SMS": true
        }
    }
}
```

Pour plus d'informations, consultez la section Gestion des numéros de téléphone des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetUserSettingsà la section Référence des AWS CLI commandes.

`get-user`

L'exemple de code suivant montre comment utiliser`get-user`.

AWS CLI

Pour obtenir des informations sur un utilisateur

L'`get-user`exemple suivant permet de récupérer les informations relatives à l'utilisateur spécifié.

```
aws chime get-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Sortie :

```
{
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "PrimaryEmail": "marthar@example.com",
```

```
        "DisplayName": "Martha Rivera",
        "LicenseType": "Pro",
        "UserRegistrationStatus": "Registered",
        "RegisteredOn": "2018-12-20T18:45:25.231Z",
        "InvitedOn": "2018-12-20T18:45:25.231Z",
        "AlexaForBusinessMetadata": {
            "IsAlexaForBusinessEnabled": False,
            "AlexaForBusinessRoomArn": "null"
        },
        "PersonalPIN": "XXXXXXXXXX"
    }
}
```

Pour plus d'informations, consultez la section Gestion des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous GetUserà la section Référence des AWS CLI commandes.

`invite-users`

L'exemple de code suivant montre comment utiliser`invite-users`.

AWS CLI

Pour inviter des utilisateurs à rejoindre Amazon Chime

L'`invite-users`exemple suivant envoie un e-mail pour inviter un utilisateur à rejoindre le compte Amazon Chime spécifié.

```
aws chime invite-users \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-email-list "alejandror@example.com" "janed@example.com"
```

Sortie :

```
{
    "Invites": [
        {
            "InviteId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "Status": "Pending",
```

```
            "EmailAddress": "alejandror@example.com",
            "EmailStatus": "Sent"
        }
        {
            "InviteId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "Status": "Pending",
            "EmailAddress": "janed@example.com",
            "EmailStatus": "Sent"
        }
    ]
}
```

Pour plus d'informations, consultez la section Invitation et suspension d'utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous InviteUsersà la section Référence des AWS CLI commandes.

`list-accounts`

L'exemple de code suivant montre comment utiliser`list-accounts`.

AWS CLI

Pour obtenir une liste de comptes

L'`list-accounts`exemple suivant permet de récupérer la liste des comptes Amazon Chime présents dans le compte de l'administrateur AWS .

```
aws chime list-accounts
```

Sortie :

```
{
    "Accounts": [
        {
            "AwsAccountId": "111122223333",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "Name": "First Chime Account",
            "AccountType": "EnterpriseDirectory",
            "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
            "DefaultLicense": "Pro",
```

```
            "SupportedLicenses": [
                "Basic",
                "Pro"
            ],
            "SigninDelegateGroups": [
                {
                    "GroupName": "myGroup"
                },
            ]
        },
        {
            "AwsAccountId": "111122223333",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "Name": "Second Chime Account",
            "AccountType": "Team",
            "CreatedTimestamp": "2018-09-04T21:44:22.292Z",
            "DefaultLicense": "Pro",
            "SupportedLicenses": [
                "Basic",
                "Pro"
            ],
            "SigninDelegateGroups": [
                {
                    "GroupName": "myGroup"
                },
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion de vos comptes Amazon Chime dans le Guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListAccountsà la section Référence des AWS CLI commandes.

`list-bots`

L'exemple de code suivant montre comment utiliser`list-bots`.

AWS CLI

Pour récupérer une liste de robots

L'`list-bots`exemple suivant répertorie les robots associés au compte Amazon Chime Enterprise spécifié.

```
aws chime list-bots \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45
```

Sortie :

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
}
```

Pour plus d'informations, consultez la section Utiliser les robots de discussion avec Amazon Chime dans le manuel du développeur Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListBotsà la section Référence des AWS CLI commandes.

`list-phone-number-orders`

L'exemple de code suivant montre comment utiliser`list-phone-number-orders`.

AWS CLI

Pour répertorier les commandes par numéro de téléphone

L'`list-phone-number-orders`exemple suivant répertorie les commandes par numéro de téléphone associées au compte de l'administrateur Amazon Chime.

```
aws chime list-phone-number-orders
```

Sortie :

```
{
    "PhoneNumberOrders": [
        {
            "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
            "ProductType": "VoiceConnector",
            "Status": "Partial",
            "OrderedPhoneNumbers": [
                {
                    "E164PhoneNumber": "+12065550100",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550101",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550102",
                    "Status": "Failed"
                }
            ],
            "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
            "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
        }
        {
            "PhoneNumberOrderId": "cba54321-ed76-09f5-321g-h54i876j2109",
            "ProductType": "BusinessCalling",
            "Status": "Partial",
            "OrderedPhoneNumbers": [
                {
                    "E164PhoneNumber": "+12065550103",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550104",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550105",
                    "Status": "Failed"
                }
            ],
            "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
```

```
            "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListPhoneNumberOrdersà la section Référence des AWS CLI commandes.

`list-phone-numbers`

L'exemple de code suivant montre comment utiliser`list-phone-numbers`.

AWS CLI

Pour répertorier les numéros de téléphone d'un compte Amazon Chime

L'`list-phone-numbers`exemple suivant répertorie les numéros de téléphone associés au compte Amazon Chime de l'administrateur.

```
aws chime list-phone-numbers
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "PhoneNumbers": [
        {
            "PhoneNumberId": "%2B12065550100",
            "E164PhoneNumber": "+12065550100",
            "Type": "Local",
            "ProductType": "VoiceConnector",
            "Status": "Assigned",
            "Capabilities": {
                "InboundCall": true,
                "OutboundCall": true,
                "InboundSMS": true,
                "OutboundSMS": true,
                "InboundMMS": true,
                "OutboundMMS": true
            },
```

```
            "Associations": [
                {
                    "Value": "abcdef1ghij2klmno3pqr4",
                    "Name": "VoiceConnectorId",
                    "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
                }
            ],
            "CallingNameStatus": "UpdateInProgress",
            "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
            "UpdatedTimestamp": "2019-10-28T18:42:07.964Z"
        },
        {
            "PhoneNumberId": "%2B12065550101",
            "E164PhoneNumber": "+12065550101",
            "Type": "Local",
            "ProductType": "VoiceConnector",
            "Status": "Assigned",
            "Capabilities": {
                "InboundCall": true,
                "OutboundCall": true,
                "InboundSMS": true,
                "OutboundSMS": true,
                "InboundMMS": true,
                "OutboundMMS": true
            },
            "Associations": [
                {
                    "Value": "abcdef1ghij2klmno3pqr4",
                    "Name": "VoiceConnectorId",
                    "AssociatedTimestamp": "2019-10-28T18:40:37.511Z"
                }
            ],
            "CallingNameStatus": "UpdateInProgress",
            "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
            "UpdatedTimestamp": "2019-10-28T18:42:07.960Z"
        }
    ]
}
```


Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListPhoneNumbersà la section Référence des AWS CLI commandes.

`list-room-memberships`

L'exemple de code suivant montre comment utiliser`list-room-memberships`.

AWS CLI

Pour répertorier les abonnements aux chambres

L'`list-room-memberships`exemple suivant affiche une liste des détails d'adhésion pour le salon de discussion spécifié.

```
aws chime list-room-memberships \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Sortie :

```
{
    "RoomMemberships": [
        {
            "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
            "Member": {
                "MemberId": "2ab2345c-67de-8901-f23g-45h678901j2k",
                "MemberType": "User",
                "Email": "zhangw@example.com",
                "FullName": "Zhang Wei",
                "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
            },
            "Role": "Member",
            "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
            "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
        },
        {
            "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
            "Member": {
                "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
                "MemberType": "User",
                "Email": "janed@example.com",
                "FullName": "Jane Doe",
                "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
            },
            "Role": "Administrator",
            "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
```

```
            "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListRoomMembershipsà la section Référence des AWS CLI commandes.

`list-rooms`

L'exemple de code suivant montre comment utiliser`list-rooms`.

AWS CLI

Pour répertorier les salons de discussion

L'`list-rooms`exemple suivant affiche la liste des salons de discussion du compte spécifié. La liste est filtrée uniquement pour les salons de discussion auxquels appartient le membre spécifié.

```
aws chime list-rooms \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Sortie :

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "teamRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListRoomsà la section Référence des AWS CLI commandes.

`list-users`

L'exemple de code suivant montre comment utiliser`list-users`.

AWS CLI

Pour répertorier les utilisateurs d'un compte

L'`list-users`exemple suivant répertorie les utilisateurs du compte Amazon Chime spécifié.

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
    "Users": [
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "mariag@example.com",
            "DisplayName": "Maria Garcia",
            "LicenseType": "Pro",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:45:25.231Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        },
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "richardr@example.com",
            "DisplayName": "Richard Roe",
            "LicenseType": "Pro",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:45:45.415Z"
            "AlexaForBusinessMetadata": {
```

```
                "IsAlexaForBusinessEnabled": false
            }
        },
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "saanvis@example.com",
            "DisplayName": "Saanvi Sarkar",
            "LicenseType": "Basic",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:46:57.747Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        },
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "wxiulan@example.com",
            "DisplayName": "Wang Xiulan",
            "LicenseType": "Basic",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:47:15.390Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`logout-user`

L'exemple de code suivant montre comment utiliser`logout-user`.

AWS CLI

Pour déconnecter un utilisateur

L'`logout-user`exemple suivant déconnecte l'utilisateur spécifié.

```
aws chime logout-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous LogoutUserà la section Référence des AWS CLI commandes.

`regenerate-security-token`

L'exemple de code suivant montre comment utiliser`regenerate-security-token`.

AWS CLI

Pour régénérer un jeton de sécurité

L'`regenerate-security-token`exemple suivant régénère le jeton de sécurité pour le bot spécifié.

```
aws chime regenerate-security-token \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Sortie :

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
```

```
    }
}
```

Pour plus d'informations, consultez Authentifier les demandes de chatbot dans le manuel Amazon Chime Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RegenerateSecurityTokenà la section Référence des AWS CLI commandes.

`reset-personal-pin`

L'exemple de code suivant montre comment utiliser`reset-personal-pin`.

AWS CLI

Pour réinitialiser le code PIN de réunion personnel d'un utilisateur

L'`reset-personal-pin`exemple suivant réinitialise le code PIN de réunion personnel de l'utilisateur spécifié.

```
aws chime reset-personal-pin \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Sortie :

```
{
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "PrimaryEmail": "mateo@example.com",
        "DisplayName": "Mateo Jackson",
        "LicenseType": "Pro",
        "UserType": "PrivateUser",
        "UserRegistrationStatus": "Registered",
        "RegisteredOn": "2018-12-20T18:45:25.231Z",
        "AlexaForBusinessMetadata": {
            "IsAlexaForBusinessEnabled": False,
            "AlexaForBusinessRoomArn": "null"
        },
        "PersonalPIN": "XXXXXXXXXX"
    }
```

```
}
```

Pour plus d'informations, consultez la section Changer de réunion personnelle PINs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous ResetPersonalPinà la section Référence des AWS CLI commandes.

`restore-phone-number`

L'exemple de code suivant montre comment utiliser`restore-phone-number`.

AWS CLI

Pour restaurer un numéro de téléphone

L'`restore-phone-number`exemple suivant restaure le numéro de téléphone spécifié à partir de la file d'attente de suppression.

```
aws chime restore-phone-number \
    --phone-number-id "+12065550100"
```

Sortie :

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "BusinessCalling",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
            "OutboundMMS": true
        },
        "Associations": [],
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-12T22:06:36.355Z"
```

```
    }
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous RestorePhoneNumberà la section Référence des AWS CLI commandes.

`search-available-phone-numbers`

L'exemple de code suivant montre comment utiliser`search-available-phone-numbers`.

AWS CLI

Pour rechercher les numéros de téléphone disponibles

L'`search-available-phone-numbers`exemple suivant recherche les numéros de téléphone disponibles par code régional.

```
aws chime search-available-phone-numbers \
    --area-code "206"
```

Sortie :

```
{
    "E164PhoneNumbers": [
        "+12065550100",
        "+12065550101",
        "+12065550102",
        "+12065550103",
        "+12065550104",
        "+12065550105",
        "+12065550106",
        "+12065550107",
        "+12065550108",
        "+12065550109",
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous SearchAvailablePhoneNumbersà la section Référence des AWS CLI commandes.

`update-account-settings`

L'exemple de code suivant montre comment utiliser`update-account-settings`.

AWS CLI

Pour mettre à jour les paramètres de votre compte

L'`update-account-settings`exemple suivant désactive le contrôle à distance des écrans partagés pour le compte Amazon Chime spécifié.

```
aws chime update-account-settings \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --account-settings DisableRemoteControl=true
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateAccountSettingsà la section Référence des AWS CLI commandes.

`update-account`

L'exemple de code suivant montre comment utiliser`update-account`.

AWS CLI

Pour mettre à jour un compte

L'`update-account`exemple suivant met à jour le nom de compte spécifié.

```
aws chime update-account \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --name MyAccountName
```

Sortie :

```
{
    "Account": {
```

```
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "MyAccountName",
        "AccountType": "Team",
        "CreatedTimestamp": "2018-09-04T21:44:22.292Z",
        "DefaultLicense": "Pro",
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
                "GroupName": "myGroup"
            },
        ]
    }
}
```

Pour plus d'informations, consultez la section Modification du nom de votre compte dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdateAccountà la section Référence des AWS CLI commandes.

`update-bot`

L'exemple de code suivant montre comment utiliser`update-bot`.

AWS CLI

Pour mettre à jour un bot

L'`update-bot`exemple suivant met à jour le statut du bot spécifié pour arrêter son exécution.

```
aws chime update-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \
    --disabled
```

Sortie :

```
{
```

```
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": true,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
    }
}
```

Pour plus d'informations, consultez la section Mettre à jour les robots de discussion dans le manuel Amazon Chime Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateBotà la section Référence des AWS CLI commandes.

`update-global-settings`

L'exemple de code suivant montre comment utiliser`update-global-settings`.

AWS CLI

Pour mettre à jour les paramètres globaux

L'`update-global-settings`exemple suivant met à jour le compartiment S3 utilisé pour stocker les enregistrements détaillés des appels pour Amazon Chime Business Calling et les connecteurs vocaux Amazon Chime associés au compte de l'administrateur. AWS

```
aws chime update-global-settings \
    --business-calling CdrBucket="s3bucket" \
    --voice-connector CdrBucket="s3bucket"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des paramètres globaux dans le Guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdateGlobalSettingsà la section Référence des AWS CLI commandes.

`update-phone-number-settings`

L'exemple de code suivant montre comment utiliser`update-phone-number-settings`.

AWS CLI

Pour mettre à jour le nom d'un appel sortant

L'`update-phone-number-settings`exemple suivant met à jour le nom d'appel sortant par défaut pour le AWS compte de l'administrateur.

```
aws chime update-phone-number-settings \
    --calling-name "myName"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdatePhoneNumberSettingsà la section Référence des AWS CLI commandes.

`update-phone-number`

L'exemple de code suivant montre comment utiliser`update-phone-number`.

AWS CLI

Exemple 1 : pour mettre à jour le type de produit pour un numéro de téléphone

L'`update-phone-number`exemple suivant met à jour le type de produit du numéro de téléphone spécifié.

```
aws chime update-phone-number \
    --phone-number-id "+12065550100" \
    --product-type "BusinessCalling"
```

Sortie :

```
{
    "PhoneNumber": {
```

```
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "BusinessCalling",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
            "OutboundMMS": true
        },
        "Associations": [],
        "CallingName": "phonenumber1",
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
    }
}
```

Exemple 2 : pour mettre à jour le nom d'appel sortant d'un numéro de téléphone

L'`update-phone-number`exemple suivant met à jour le nom de l'appel sortant pour le numéro de téléphone spécifié.

as carillon update-phone-number - phone-number-id « +12065550100" --calling-name « phonenumber2 »

Sortie :

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "BusinessCalling",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
```

```
            "OutboundMMS": true
        },
        "Associations": [],
        "CallingName": "phonenumber2",
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des numéros de téléphone dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdatePhoneNumberà la section Référence des AWS CLI commandes.

`update-room-membership`

L'exemple de code suivant montre comment utiliser`update-room-membership`.

AWS CLI

Pour mettre à jour l'abonnement à une chambre

L'`update-room-membership`exemple suivant modifie le rôle du membre du salon de discussion spécifié en`Administrator`.

```
aws chime update-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k \
    --role Administrator
```

Sortie :

```
{
    "RoomMembership": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Member": {
            "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
            "MemberType": "User",
            "Email": "sofiamartinez@example.com",
            "FullName": "Sofia Martinez",
```

```
            "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
        },
        "Role": "Administrator",
        "InvitedBy": "arn:aws:iam::111122223333:user/admin",
        "UpdatedTimestamp": "2019-12-02T22:40:22.931Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdateRoomMembershipà la section Référence des AWS CLI commandes.

`update-room`

L'exemple de code suivant montre comment utiliser`update-room`.

AWS CLI

Pour mettre à jour un salon de discussion

L'`update-room`exemple suivant modifie le nom du salon de discussion spécifié.

```
aws chime update-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --name teamRoom
```

Sortie :

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "teamRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un salon de discussion dans le guide de l'utilisateur d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdateRoomà la section Référence des AWS CLI commandes.

`update-user-settings`

L'exemple de code suivant montre comment utiliser`update-user-settings`.

AWS CLI

Pour mettre à jour les paramètres utilisateur

L'`update-user-settings`exemple suivant permet à l'utilisateur spécifié de passer des appels entrants et sortants et d'envoyer et de recevoir des SMS.

```
aws chime update-user-settings \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \
    --user-settings "Telephony={InboundCalling=true,OutboundCalling=true,SMS=true}"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des numéros de téléphone des utilisateurs dans le guide d'administration d'Amazon Chime.

- Pour plus de détails sur l'API, reportez-vous UpdateUserSettingsà la section Référence des AWS CLI commandes.

`update-user`

L'exemple de code suivant montre comment utiliser`update-user`.

AWS CLI

Pour mettre à jour les informations de l'utilisateur

Cet exemple met à jour les informations spécifiées pour l'utilisateur spécifié.

Commande :

```
aws chime update-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \
    --license-type "Basic"
```

Sortie :

```
{
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateUserà la section Référence des AWS CLI commandes.

Exemples d'API Cloud Control utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l'API AWS Command Line Interface with Cloud Control.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-resource`

L'exemple de code suivant montre comment utiliser`create-resource`.

AWS CLI

Pour créer une ressource

L'`create-resource`exemple suivant crée une ressource AWS : :Kinesis : :Stream, nommée ResourceExample, avec une période de rétention de 168 heures et un nombre de partitions de trois.

```
aws cloudcontrol create-resource \
    --type-name AWS::Kinesis::Stream \
    --desired-state "{\"Name\": \"ResourceExample\",\"RetentionPeriodHours\":168,
 \"ShardCount\":3}"
```

Sortie :

```
{
    "ProgressEvent": {
        "EventTime": 1632506656.706,
        "TypeName": "AWS::Kinesis::Stream",
        "OperationStatus": "IN_PROGRESS",
        "Operation": "CREATE",
        "Identifier": "ResourceExample",
        "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"
    }
}
```

Pour plus d'informations, consultez la section Création d'une ressource dans le guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous CreateResourceà la section Référence des AWS CLI commandes.

`delete-resource`

L'exemple de code suivant montre comment utiliser`delete-resource`.

AWS CLI

Pour supprimer une ressource

L'`delete-resource`exemple suivant supprime une ressource AWS : :Kinesis : :Stream avec l'identifiant ResourceExample de votre compte. AWS

```
aws cloudcontrol delete-resource \
    --type-name AWS::Kinesis::Stream \
```

```
    --identifier ResourceExample
```

Sortie :

```
{
    "ProgressEvent": {
        "TypeName": "AWS::Kinesis::Stream",
        "Identifier": "ResourceExample",
        "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",
        "Operation": "DELETE",
        "OperationStatus": "IN_PROGRESS",
        "EventTime": 1632950300.14
    }
}
```

Pour plus d'informations, consultez la section Suppression d'une ressource dans le guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous DeleteResourceà la section Référence des AWS CLI commandes.

`get-resource-request-status`

L'exemple de code suivant montre comment utiliser`get-resource-request-status`.

AWS CLI

Pour obtenir les informations d'état d'une demande de ressource

L'`get-resource-request-status`exemple suivant renvoie des informations d'état relatives à la demande de ressource spécifiée.

```
aws cloudcontrol get-resource-request-status \
    --request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

Sortie :

```
{
    "ProgressEvent": {
        "TypeName": "AWS::Kinesis::Stream",
        "Identifier": "Demo",
```

```
        "RequestToken": "e1a6b86e-46bd-41ac-bfba-001234567890",
        "Operation": "CREATE",
        "OperationStatus": "FAILED",
        "EventTime": 1632950268.481,
        "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with identifier
 'Demo' already exists.",
        "ErrorCode": "AlreadyExists"
    }
}
```

Pour plus d'informations, consultez la section Gestion des demandes d'opérations sur les ressources dans le Guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous GetResourceRequestStatusà la section Référence des AWS CLI commandes.

`get-resource`

L'exemple de code suivant montre comment utiliser`get-resource`.

AWS CLI

Pour obtenir l'état actuel d'une ressource

L'`get-resource`exemple suivant renvoie l'état actuel de la ressource AWS : :Kinesis : :Stream nommée. ResourceExample

```
aws cloudcontrol get-resource \
    --type-name AWS::Kinesis::Stream \
    --identifier ResourceExample
```

Sortie :

```
{
    "TypeName": "AWS::Kinesis::Stream",
    "ResourceDescription": {
        "Identifier": "ResourceExample",
        "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/
ResourceExample\",\"RetentionPeriodHours\":168,\"Name\":\"ResourceExample\",
\"ShardCount\":3}"
    }
}
```

Pour plus d'informations, consultez la section Lire l'état actuel d'une ressource dans le Guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous GetResourceà la section Référence des AWS CLI commandes.

`list-resource-requests`

L'exemple de code suivant montre comment utiliser`list-resource-requests`.

AWS CLI

Pour répertorier les demandes d'opérations de ressources actives

L'`list-resource-requests`exemple suivant répertorie les demandes de ressources pour les opérations CREATE et UPDATE qui ont échoué dans votre AWS compte.

```
aws cloudcontrol list-resource-requests \
    --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED
```

Sortie :

```
{
    "ResourceRequestStatusSummaries": [
        {
            "TypeName": "AWS::Kinesis::Stream",
            "Identifier": "Demo",
            "RequestToken": "e1a6b86e-46bd-41ac-bfba-633abcdfdbd7",
            "Operation": "CREATE",
            "OperationStatus": "FAILED",
            "EventTime": 1632950268.481,
            "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with
 identifier 'Demo' already exists.",
            "ErrorCode": "AlreadyExists"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des demandes d'opérations sur les ressources dans le Guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous ListResourceRequestsà la section Référence des AWS CLI commandes.

`list-resources`

L'exemple de code suivant montre comment utiliser`list-resources`.

AWS CLI

Pour répertorier les ressources d'un type donné

L'`list-resources`exemple suivant répertorie les ressources ::Kinesis AWS : :Stream mises en service dans votre compte. AWS

```
aws cloudcontrol list-resources \
    --type-name AWS::Kinesis::Stream
```

Sortie :

```
{
    "TypeName": "AWS::Kinesis::Stream",
    "ResourceDescriptions": [
        {
            "Identifier": "MyKinesisStream",
            "Properties": "{\"Name\":\"MyKinesisStream\"}"
        },
        {
            "Identifier": "AnotherStream",
            "Properties": "{\"Name\":\"AnotherStream\"}"
        }
    ]
}
```

Pour plus d'informations, consultez la section Découverte des ressources dans le guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous ListResourcesà la section Référence des AWS CLI commandes.

`update-resource`

L'exemple de code suivant montre comment utiliser`update-resource`.

AWS CLI

Pour mettre à jour les propriétés d'une ressource existante

L'`update-resource`exemple suivant met à jour la politique de rétention d'une LogGroup ressource AWS : :Logs : : nommée ExampleLogGroup à 90 jours.

```
aws cloudcontrol update-resource \
    --type-name AWS::Logs::LogGroup \
    --identifier ExampleLogGroup \
    --patch-document "[{\"op\":\"replace\",\"path\":\"/RetentionInDays\",\"value
\":90}]"
```

Sortie :

```
{
    "ProgressEvent": {
        "EventTime": "2021-08-09T18:17:15.219Z",
        "TypeName": "AWS::Logs::LogGroup",
        "OperationStatus": "IN_PROGRESS",
        "Operation": "UPDATE",
        "Identifier": "ExampleLogGroup",
        "RequestToken": "5f40c577-3534-4b20-9599-0b0123456789"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une ressource dans le guide de l'utilisateur de l'API Cloud Control.

- Pour plus de détails sur l'API, reportez-vous UpdateResourceà la section Référence des AWS CLI commandes.

AWS Cloud Map exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Cloud Map.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-http-namespace`

L'exemple de code suivant montre comment utiliser`create-http-namespace`.

AWS CLI

Pour créer un espace de noms HTTP

L'`create-http-namespace`exemple suivant crée un espace de noms `example.com` HTTP.

```
aws servicediscovery create-http-namespace \
    --name example.com \
    --creator-request-id example-request-id
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`. Pour plus d'informations, consultez get-operation.

Pour plus d'informations sur la création d'un espace de noms, consultez la section Création d'un espace de noms AWS Cloud Map pour regrouper les services d'application dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir CreateHttpNamespacela section Référence des AWS CLI commandes.

`create-private-dns-namespace`

L'exemple de code suivant montre comment utiliser`create-private-dns-namespace`.

AWS CLI

Pour créer un espace de noms DNS privé

L'`create-private-dns-namespace`exemple suivant crée un espace de noms DNS privé.

```
aws servicediscovery create-private-dns-namespace \
    --name example.com \
    --vpc vpc-1c56417b
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`. Pour plus d'informations, consultez get-operation.

Pour plus d'informations, consultez la section Création d'espaces de noms dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir CreatePrivateDnsNamespacela section Référence des AWS CLI commandes.

`create-public-dns-namespace`

L'exemple de code suivant montre comment utiliser`create-public-dns-namespace`.

AWS CLI

Pour créer un espace de noms DNS public

L'`create-public-dns-namespace`exemple suivant crée un espace de noms `example.com` DNS public.

```
aws servicediscovery create-public-dns-namespace \
    --name example-public-dns.com \
    --creator-request-id example-public-request-id \
```

```
    --properties DnsProperties={SOA={TTL=60}}
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`.

Pour plus d'informations sur la création d'un espace de noms, consultez la section Création d'un espace de noms AWS Cloud Map pour regrouper les services d'application dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir CreatePublicDnsNamespacela section Référence des AWS CLI commandes.

`create-service`

L'exemple de code suivant montre comment utiliser`create-service`.

AWS CLI

Pour créer un service

L'`create-service`exemple suivant crée un service.

```
aws servicediscovery create-service \
    --name myservice \
    --namespace-id  ns-ylexjili4cdxy3xm \
    --dns-config "NamespaceId=ns-
ylexjili4cdxy3xm,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

Sortie :

```
{
        "Service": {
        "Id": "srv-p5zdwlg5uvvzjita",
        "Arn": "arn:aws:servicediscovery:us-west-2:803642222207:service/srv-
p5zdwlg5uvvzjita",
        "Name": "myservice",
        "NamespaceId": "ns-ylexjili4cdxy3xm",
```

```
        "DnsConfig": {
            "NamespaceId": "ns-ylexjili4cdxy3xm",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 60
                }
            ]
        },
        "CreateDate": 1587081768.334,
        "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"
    }
}
```

Pour plus d'informations, consultez la section Création de services dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir CreateServicela section Référence des AWS CLI commandes.

`delete-namespace`

L'exemple de code suivant montre comment utiliser`delete-namespace`.

AWS CLI

Pour supprimer un espace de noms

L'`delete-namespace`exemple suivant supprime un espace de noms.

```
aws servicediscovery delete-namespace \
    --id ns-ylexjili4cdxy3xm
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`. Pour plus d'informations, consultez get-operation.

Pour plus d'informations, consultez la section Supprimer des espaces de noms dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir DeleteNamespacela section Référence des AWS CLI commandes.

`delete-service-attributes`

L'exemple de code suivant montre comment utiliser`delete-service-attributes`.

AWS CLI

Pour supprimer un attribut de service

L'`delete-service-attributes`exemple suivant supprime un attribut de service avec la clé `Port` associée au service spécifié.

```
aws servicediscovery delete-service-attributes \
    --service-id srv-e4anhexample0004 \
    --attributes Port
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Supprimer des espaces de noms dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir DeleteServiceAttributesla section Référence des AWS CLI commandes.

`delete-service`

L'exemple de code suivant montre comment utiliser`delete-service`.

AWS CLI

Pour supprimer un service

L'`delete-service`exemple suivant supprime un service.

```
aws servicediscovery delete-service \
    --id srv-p5zdwlg5uvvzjita
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression de services dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir DeleteServicela section Référence des AWS CLI commandes.

`deregister-instance`

L'exemple de code suivant montre comment utiliser`deregister-instance`.

AWS CLI

Pour désenregistrer une instance de service

L'`deregister-instance`exemple suivant annule l'enregistrement d'une instance de service.

```
aws servicediscovery deregister-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53
```

Sortie :

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k98rnaiq"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`. Pour plus d'informations, consultez get-operation.

Pour plus d'informations, consultez la section Désenregistrement des instances de service dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir DeregisterInstancela section Référence des AWS CLI commandes.

`discover-instances-revision`

L'exemple de code suivant montre comment utiliser`discover-instances-revision`.

AWS CLI

Pour découvrir la révision d'une instance

L'`discover-instances-revision`exemple suivant découvre la révision croissante d'une instance.

```
aws servicediscovery discover-instances-revision \
    --namespace-name example.com \
    --service-name myservice
```

Sortie :

```
{
    "InstancesRevision": 123456
}
```

Pour plus d'informations, consultez la section Instances du service AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir DiscoverInstancesRevisionla section Référence des AWS CLI commandes.

`discover-instances`

L'exemple de code suivant montre comment utiliser`discover-instances`.

AWS CLI

Pour découvrir les instances enregistrées

L'`discover-instances`exemple suivant découvre les instances enregistrées.

```
aws servicediscovery discover-instances \
    --namespace-name example.com \
    --service-name myservice \
    --max-results 10 \
    --health-status ALL
```

Sortie :

```
{
    "Instances": [
        {
            "InstanceId": "myservice-53",
            "NamespaceName": "example.com",
            "ServiceName": "myservice",
            "HealthStatus": "UNKNOWN",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.2.1.3",
                "AWS_INSTANCE_PORT": "808"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Instances du service AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir DiscoverInstancesla section Référence des AWS CLI commandes.

`get-instance`

L'exemple de code suivant montre comment utiliser`get-instance`.

AWS CLI

Pour obtenir les détails d'une instance

L'`get-instance`exemple suivant permet d'obtenir les attributs d'un service.

```
aws servicediscovery get-instance \
    --service-id srv-e4anhexample0004
    --instance-id i-abcd1234
```

Sortie :

```
{
    "Instances": {
        "Id": "arn:aws:servicediscovery:us-west-2:111122223333;:service/srv-
e4anhexample0004",
        "Attributes": {
```

```
            "AWS_INSTANCE_IPV4": "192.0.2.44",
            "AWS_INSTANCE_PORT": "80",
            "color": "green",
            "region": "us-west-2",
            "stage": "beta"
        }
    }
}
```

Pour plus d'informations, consultez la section Instances du service AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir GetInstancela section Référence des AWS CLI commandes.

`get-instances-health-status`

L'exemple de code suivant montre comment utiliser`get-instances-health-status`.

AWS CLI

Pour obtenir l'état de santé des instances associées à un service

L'`get-instances-health-status`exemple suivant permet d'obtenir l'état de santé des instances associées au service spécifié.

```
aws servicediscovery get-instances-health-status \
    --service-id srv-e4anhexample0004
```

Sortie :

```
{
    "Status": {
        "i-abcd1234": "HEALTHY",
        "i-abcd1235": "UNHEALTHY"
    }
}
```

Pour plus d'informations, consultez la section Instances du service AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir GetInstancesHealthStatusla section Référence des AWS CLI commandes.

`get-namespace`

L'exemple de code suivant montre comment utiliser`get-namespace`.

AWS CLI

Pour obtenir les détails d'un espace de noms

L'`get-namespace`exemple suivant récupère des informations sur l'espace de noms spécifié.

```
aws servicediscovery get-namespace \
    --id ns-e4anhexample0004
```

Sortie :

```
{
    "Namespaces": {
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004",
        "CreateDate": "20181118T211712Z",
        "CreatorRequestId": "example-creator-request-id-0001",
        "Description": "Example.com AWS Cloud Map HTTP Namespace",
        "Id": "ns-e4anhexample0004",
        "Name": "example-http.com",
        "Properties": {
            "DnsProperties": {},
            "HttpProperties": {
                "HttpName": "example-http.com"
            }
        },
        "Type": "HTTP"
    }
}
```

Pour plus d'informations, consultez la section Espaces de noms AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir GetNamespacela section Référence des AWS CLI commandes.

`get-operation`

L'exemple de code suivant montre comment utiliser`get-operation`.

AWS CLI

Pour obtenir le résultat d'une opération

L'`get-operation`exemple suivant obtient le résultat d'une opération de création d'espace de noms.

```
aws servicediscovery get-operation \
    --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

Sortie :

```
{
    "Operation": {
        "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",
        "Type": "CREATE_NAMESPACE",
        "Status": "SUCCESS",
        "CreateDate": 1587055860.121,
        "UpdateDate": 1587055900.469,
        "Targets": {
            "NAMESPACE": "ns-ylexjili4cdxy3xm"
        }
    }
}
```

Pour plus d'informations, consultez la section Création d'un espace de noms AWS Cloud Map pour regrouper les services d'application dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir GetOperationla section Référence des AWS CLI commandes.

`get-service-attributes`

L'exemple de code suivant montre comment utiliser`get-service-attributes`.

AWS CLI

Pour obtenir les attributs d'un service

L'`get-service-attributes`exemple suivant permet d'obtenir les attributs d'un service.

```
aws servicediscovery get-service-attributes \
```

```
    --service-id srv-e4anhexample0004
```

Sortie :

```
{
    "ServiceAttributes": {
        "ServiceArn": "arn:aws:servicediscovery:us-west-2:111122223333;:service/srv-
e4anhexample0004",
        "Attributes": {
            "Port": "80"
        }
    }
}
```

Pour plus d'informations, consultez la section Services AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir GetServiceAttributesla section Référence des AWS CLI commandes.

`get-service`

L'exemple de code suivant montre comment utiliser`get-service`.

AWS CLI

Pour obtenir les paramètres d'un service

L'`get-service`exemple suivant permet d'obtenir les paramètres d'un service spécifié.

```
aws servicediscovery get-service \
    --id srv-e4anhexample0004
```

Sortie :

```
{
    "Service": {
        "Id": "srv-e4anhexample0004",
        "Arn": "arn:aws:servicediscovery:us-west-2:111122223333:service/srv-
e4anhexample0004",
        "Name": "test-service",
        "NamespaceId": "ns-e4anhexample0004",
```

```
        "DnsConfig": {},
        "Type": "HTTP",
        "CreateDate": "2025-02-24T10:59:02.905000-06:00",
        "CreatorRequestId": "3f50f9d9-b14c-482e-a556-d2a22fe6106d"
    }
}
```

Pour plus d'informations, consultez la section Services AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir GetServicela section Référence des AWS CLI commandes.

`list-instances`

L'exemple de code suivant montre comment utiliser`list-instances`.

AWS CLI

Pour répertorier les instances de service

L'`list-instances`exemple suivant répertorie les instances de service.

```
aws servicediscovery list-instances \
    --service-id srv-qzpwvt2tfqcegapy
```

Sortie :

```
{
    "Instances": [
        {
            "Id": "i-06bdabbae60f65a4e",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.2.1.3",
                "AWS_INSTANCE_PORT": "808"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste d'instances de service dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir ListInstancesla section Référence des AWS CLI commandes.

`list-namespaces`

L'exemple de code suivant montre comment utiliser`list-namespaces`.

AWS CLI

Pour répertorier les espaces de noms

L'`list-namespaces`exemple suivant répertorie les espaces de noms.

```
aws servicediscovery list-namespaces
```

Sortie :

```
{
    "Namespaces": [
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
a3ccy2e7e3a7rile",
            "CreateDate": 1585354387.357,
            "Id": "ns-a3ccy2e7e3a7rile",
            "Name": "local",
            "Properties": {
                "DnsProperties": {
                    "HostedZoneId": "Z06752353VBUDTC32S84S"
                },
                "HttpProperties": {
                    "HttpName": "local"
                 }
            },
            "Type": "DNS_PRIVATE"
        },
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
pocfyjtrsmwtvcxx",
            "CreateDate": 1586468974.698,
            "Description": "My second namespace",
            "Id": "ns-pocfyjtrsmwtvcxx",
            "Name": "My-second-namespace",
            "Properties": {
                "DnsProperties": {},
                "HttpProperties": {
                    "HttpName": "My-second-namespace"
                }
```

```
            },
            "Type": "HTTP"
        },
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
ylexjili4cdxy3xm",
            "CreateDate": 1587055896.798,
            "Id": "ns-ylexjili4cdxy3xm",
            "Name": "example.com",
            "Properties": {
                "DnsProperties": {
                    "HostedZoneId": "Z09983722P0QME1B3KC8I"
                },
                 "HttpProperties": {
                     "HttpName": "example.com"
                }
            },
            "Type": "DNS_PRIVATE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste d'espaces de noms dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir ListNamespaceslа section Référence des AWS CLI commandes.

`list-operations`

L'exemple de code suivant montre comment utiliser`list-operations`.

AWS CLI

Pour répertorier les opérations répondant aux critères spécifiés

L'`list-operations`exemple suivant répertorie les opérations dont le statut est `PENDING` ou`SUCCESS`.

```
aws servicediscovery list-operations \
    --service-id srv-e4anhexample0004 \
    --filters Name=STATUS,Condition=IN,Values=PENDING,SUCCESS
```

Sortie :

```
{
    "Operations": [
        {
            "Id": "76yy8ovhpdz0plmjzbsnqgnrqvpv2qdt-kexample",
            "Status": "SUCCESS"
        },
        {
            "Id": "prysnyzpji3u2ciy45nke83x2zanl7yk-dexample",
            "Status": "SUCCESS"
        },
        {
            "Id": "ko4ekftir7kzlbechsh7xvcdgcpk66gh-7example",
            "Status": "PENDING"
        }
    ]
}
```

Pour plus d'informations, consultez Qu'est-ce que AWS Cloud Map ? dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir ListOperationsla section Référence des AWS CLI commandes.

`list-services`

L'exemple de code suivant montre comment utiliser`list-services`.

AWS CLI

Pour répertorier les services

L'`list-services`exemple suivant répertorie les services.

```
aws servicediscovery list-services
```

Sortie :

```
{
    "Services": [
        {
```

```
            "Id": "srv-p5zdwlg5uvvzjita",
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
p5zdwlg5uvvzjita",
            "Name": "myservice",
            "DnsConfig": {
                "RoutingPolicy": "MULTIVALUE",
                "DnsRecords": [
                    {
                        "Type": "A",
                        "TTL": 60
                    }
                ]
            },
            "CreateDate": 1587081768.334
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste de services dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir ListServicesla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises associées à la ressource spécifiée

L'`list-tags-for-resource`exemple suivant répertorie les balises pour la ressource spécifiée.

```
aws servicediscovery list-tags-for-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004
```

Sortie :

```
{
    "Tags": [
        {
```

```
            "Key": "Project",
            "Value": "Zeta"
        },
        {
            "Key": "Department",
            "Value": "Engineering"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage de vos ressources AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`register-instance`

L'exemple de code suivant montre comment utiliser`register-instance`.

AWS CLI

Pour enregistrer une instance de service

L'`register-instance`exemple suivant enregistre une instance de service.

```
aws servicediscovery register-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53 \
    --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

Sortie :

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k95yg2u7"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`. Pour plus d'informations, consultez get-operation.

Pour plus d'informations, consultez la section Enregistrement des instances dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir RegisterInstancela section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour associer des balises à la ressource spécifiée

L'`tag-resource`exemple suivant associe une `Department` balise à la valeur `Engineering` associée à l'espace de noms spécifié.

```
aws servicediscovery tag-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-e4anhexample0004 \
    --tags Key=Department, Value=Engineering
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises de la ressource spécifiée

L'`untag-resource`exemple suivant supprime une `Department` balise de l'espace de noms spécifié.

```
aws servicediscovery untag-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-e4anhexample0004 \
    --tags Key=Department, Value=Engineering
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-http-namespace`

L'exemple de code suivant montre comment utiliser`update-http-namespace`.

AWS CLI

Pour mettre à jour un espace de noms HTTP

L'`update-http-namespace`exemple suivant met à jour la description de l'espace de noms HTTP spécifié.

```
aws servicediscovery update-http-namespace \
    --id ns-vh4nbmEXAMPLE \
    --updater-request-id example-request-id \
    --namespace Description="The updated namespace description."
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`. Pour plus d'informations, consultez get-operation.

Pour plus d'informations, consultez la section Espaces de noms AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, voir UpdateHttpNamespacela section Référence des AWS CLI commandes.

`update-instance-custom-health-status`

L'exemple de code suivant montre comment utiliser`update-instance-custom-health-status`.

AWS CLI

Pour mettre à jour un bilan de santé personnalisé

L'`update-instance-custom-health-status`exemple suivant met à jour l'état du bilan de santé personnalisé pour le service spécifié et l'exemple d'instance de service sur`HEALTHY`.

```
aws servicediscovery update-instance-custom-health-status \
    --service-id srv-e4anhexample0004 \
    --instance-id example \
    --status HEALTHY
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez AWS la section Configuration de la vérification de l'état du service AWS Cloud Map dans le Guide du développeur de Cloud Map.

- Pour plus de détails sur l'API, reportez-vous UpdateInstanceCustomHealthStatusà la section Référence des AWS CLI commandes.

`update-private-dns-namespace`

L'exemple de code suivant montre comment utiliser`update-private-dns-namespace`.

AWS CLI

Pour mettre à jour un espace de noms DNS privé

L'`update-private-dns-namespace`exemple suivant met à jour la description d'un espace de noms DNS privé.

```
aws servicediscovery update-private-dns-namespace \
    --id ns-bk3aEXAMPLE \
    --updater-request-id example-private-request-id \
    --namespace Description="The updated namespace description."
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`.

Pour plus d'informations, consultez la section Espaces de noms AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, reportez-vous UpdatePrivateDnsNamespaceà la section Référence des AWS CLI commandes.

`update-public-dns-namespace`

L'exemple de code suivant montre comment utiliser`update-public-dns-namespace`.

AWS CLI

Pour mettre à jour un espace de noms DNS public

L'`update-public-dns-namespace`exemple suivant met à jour la description d'un espace de noms DNS public.

```
aws servicediscovery update-public-dns-namespace \
    --id ns-bk3aEXAMPLE \
    --updater-request-id example-public-request-id \
    --namespace Description="The updated namespace description."
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`.

Pour plus d'informations, consultez la section Espaces de noms AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, reportez-vous UpdatePublicDnsNamespaceà la section Référence des AWS CLI commandes.

`update-service-attributes`

L'exemple de code suivant montre comment utiliser`update-service-attributes`.

AWS CLI

Pour mettre à jour un service afin d'ajouter un attribut

L'`update-service-attributes`exemple suivant met à jour le service spécifié pour ajouter un attribut de service avec une clé `Port` et une valeur`80`.

```
aws servicediscovery update-service-attributes \
    --service-id srv-e4anhexample0004 \
    --attributes Port=80
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Services AWS Cloud Map dans le Guide du développeur de AWS Cloud Map.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceAttributesà la section Référence des AWS CLI commandes.

`update-service`

L'exemple de code suivant montre comment utiliser`update-service`.

AWS CLI

Pour mettre à jour un service

L'`update-service`exemple suivant met à jour un service pour mettre à jour les `HealthCheckConfig` paramètres `DnsConfig` et.

```
aws servicediscovery update-service \
    --id srv-e4anhexample0004 \
    --service
 "DnsConfig={DnsRecords=[{"Type"="A","TTL"=60}]},HealthCheckConfig={"Type"="HTTP","ResourceP
```

Sortie :

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation`.

Pour plus d'informations sur la mise à jour d'un service, consultez la section Mise à jour d'un service AWSAWS Cloud Map dans le Guide du développeur de Cloud Map.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceà la section Référence des AWS CLI commandes.

AWS Cloud9 exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Cloud9.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-environment-ec2`

L'exemple de code suivant montre comment utiliser`create-environment-ec2`.

AWS CLI

Pour créer un environnement de AWS développement Cloud9 EC2

L'`create-environment-ec2`exemple suivant crée un environnement de développement AWS Cloud9 avec les paramètres spécifiés, lance une instance Amazon Elastic Compute Cloud EC2 (Amazon), puis connecte l'instance à l'environnement.

```
aws cloud9 create-environment-ec2 \
    --name my-demo-env \
    --description "My demonstration development environment." \
    --instance-type t2.micro --image-id amazonlinux-2023-x86_64 \
    --subnet-id subnet-1fab8aEX \
    --automatic-stop-time-minutes 60 \
```

```
    --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

Sortie :

```
{
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"
}
```

Pour plus d'informations, consultez le guide de l'utilisateur de Creating an EC2 Environment in the AWS Cloud9.

- Pour plus de détails sur l'API, reportez-vous à la section CreateEnvironmentEc2 de AWS CLI la section Référence des commandes.

`create-environment-membership`

L'exemple de code suivant montre comment utiliser`create-environment-membership`.

AWS CLI

Pour ajouter un membre d'environnement à un environnement de AWS développement Cloud9

Cet exemple ajoute le membre d'environnement spécifié à l'environnement de développement AWS Cloud9 spécifié.

Commande :

```
aws cloud9 create-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser --permissions read-write
```

Sortie :

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUXTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-write"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateEnvironmentMembershipà la section Référence des AWS CLI commandes.

`delete-environment-membership`

L'exemple de code suivant montre comment utiliser`delete-environment-membership`.

AWS CLI

Pour supprimer un membre de l'environnement d'un environnement de AWS développement Cloud9

Cet exemple supprime le membre d'environnement spécifié de l'environnement de développement AWS Cloud9 spécifié.

Commande :

```
aws cloud9 delete-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous DeleteEnvironmentMembershipà la section Référence des AWS CLI commandes.

`delete-environment`

L'exemple de code suivant montre comment utiliser`delete-environment`.

AWS CLI

Pour supprimer un environnement de AWS développement Cloud9

Cet exemple supprime l'environnement de développement AWS Cloud9 spécifié. Si une EC2 instance Amazon est connectée à l'environnement, elle met également fin à l'instance.

Commande :

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous DeleteEnvironmentà la section Référence des AWS CLI commandes.

`describe-environment-memberships`

L'exemple de code suivant montre comment utiliser`describe-environment-memberships`.

AWS CLI

Pour obtenir des informations sur les membres de l'environnement d'un environnement de AWS développement Cloud9

Cet exemple permet d'obtenir des informations sur les membres de l'environnement de développement AWS Cloud9 spécifié.

Commande :

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX
```

Sortie :

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJ3LOROMOUXTBSU6EX",
      "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
      "permissions": "read-write"
    },
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
```

```
      "permissions": "owner"
    }
  ]
}
```

Pour obtenir des informations sur le propriétaire d'un environnement de AWS développement Cloud9

Cet exemple permet d'obtenir des informations sur le propriétaire de l'environnement de développement AWS Cloud9 spécifié.

Commande :

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

Sortie :

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

Pour obtenir des informations sur un membre d'environnement pour plusieurs environnements de AWS développement Cloud9

Cet exemple permet d'obtenir des informations sur le membre d'environnement spécifié pour plusieurs environnements de développement AWS Cloud9.

Commande :

```
aws cloud9 describe-environment-memberships --user-
arn arn:aws:iam::123456789012:user/MyDemoUser
```

Sortie :

```
{
  "memberships": [
    {
      "environmentId": "10a75714bd494714929e7f5ec4125aEX",
      "lastAccess": 1516213427.0,
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    },
    {
      "environmentId": "1980b80e5f584920801c09086667f0EX",
      "lastAccess": 1516144884.0,
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEnvironmentMembershipsà la section Référence des AWS CLI commandes.

`describe-environment-status`

L'exemple de code suivant montre comment utiliser`describe-environment-status`.

AWS CLI

Pour obtenir des informations sur l'état d'un environnement de AWS développement Cloud9

Cet exemple permet d'obtenir des informations d'état pour l'environnement de développement AWS Cloud9 spécifié.

Commande :

```
aws cloud9 describe-environment-status --environment-
id 685f892f431b45c2b28cb69eadcdb0EX
```

Sortie :

```
{
```

```
  "status": "ready",
  "message": "Environment is ready to use"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEnvironmentStatusà la section Référence des AWS CLI commandes.

`describe-environments`

L'exemple de code suivant montre comment utiliser`describe-environments`.

AWS CLI

Pour obtenir des informations sur les environnements de AWS développement Cloud9

Cet exemple permet d'obtenir des informations sur les environnements de développement AWS Cloud9 spécifiés.

Commande :

```
aws cloud9 describe-environments --environment-
ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

Sortie :

```
{
  "environments": [
    {
      "id": "685f892f431b45c2b28cb69eadcdb0EX",
      "name": "my-demo-ec2-env",
      "description": "Created from CodeStar.",
      "type": "ec2",
      "arn": "arn:aws:cloud9:us-
east-1:123456789012:environment:685f892f431b45c2b28cb69eadcdb0EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    },
    {
      "id": "349c86d4579e4e7298d500ff57a6b2EX",
      "name": my-demo-ssh-env",
```

```
      "description": "",
      "type": "ssh",
      "arn": "arn:aws:cloud9:us-
east-1:123456789012:environment:349c86d4579e4e7298d500ff57a6b2EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEnvironmentsà la section Référence des AWS CLI commandes.

`list-environments`

L'exemple de code suivant montre comment utiliser`list-environments`.

AWS CLI

Pour obtenir la liste des identifiants d'environnement de développement AWS Cloud9 disponibles

Cet exemple permet d'obtenir une liste des identifiants d'environnement de développement AWS Cloud9 disponibles.

Commande :

```
aws cloud9 list-environments
```

Sortie :

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListEnvironmentsà la section Référence des AWS CLI commandes.

`update-environment-membership`

L'exemple de code suivant montre comment utiliser`update-environment-membership`.

AWS CLI

Pour modifier les paramètres d'un membre d'environnement existant pour un environnement de AWS développement Cloud9

Cet exemple modifie les paramètres du membre d'environnement existant spécifié pour l'environnement de développement AWS Cloud9 spécifié.

Commande :

```
aws cloud9 update-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser --permissions read-only
```

Sortie :

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUXTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateEnvironmentMembershipà la section Référence des AWS CLI commandes.

`update-environment`

L'exemple de code suivant montre comment utiliser`update-environment`.

AWS CLI

Pour modifier les paramètres d'un environnement de développement AWS Cloud9 existant

Cet exemple modifie les paramètres spécifiés de l'environnement de développement AWS Cloud9 existant spécifié.

Commande :

```
aws cloud9 update-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
 --name my-changed-demo-env --description "My changed demonstration development
 environment."
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous UpdateEnvironmentà la section Référence des AWS CLI commandes.

AWS CloudFormation exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS CloudFormation.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`activate-type`

L'exemple de code suivant montre comment utiliser`activate-type`.

AWS CLI

Pour activer un type

L'`activate-type`exemple suivant active une extension tierce publique, la rendant disponible pour une utilisation dans les modèles de pile.

```
aws cloudformation activate-type \
    --region us-west-2 \
    --type RESOURCE \
    --type-name Example::Test::1234567890abcdef0 \
    --type-name-alias Example::Test::Alias
```

Sortie :

```
{
    "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-
Test-Alias"
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ActivateTypeà la section Référence des AWS CLI commandes.

`batch-describe-type-configurations`

L'exemple de code suivant montre comment utiliser`batch-describe-type-configurations`.

AWS CLI

Pour décrire par lots une configuration de type

L'`batch-describe-type-configurations`exemple suivant configure les données pour le type.

```
aws cloudformation batch-describe-type-configurations \
    --region us-west-2 \
    --type-configuration-identifiers TypeArn="arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Example-Test-
Type,TypeConfigurationAlias=MyConfiguration"
```

Sortie :

```
{
    "Errors": [],
```

```
    "UnprocessedTypeConfigurations": [],
    "TypeConfigurations": [
        {
            "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
Example-Test-Type",
            "Alias": "MyConfiguration",
            "Configuration": "{\n        \"Example\": {\n            \"ApiKey\":
 \"examplekey\",\n            \"ApplicationKey\": \"examplekey1\",\n
 \"ApiURL\": \"exampleurl\"\n            }\n}",
            "LastUpdated": "2021-10-01T15:25:46.210000+00:00",
            "TypeArn": "arn:aws:cloudformation:us-east-1:123456789012:type/resource/
Example-Test-Type"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchDescribeTypeConfigurationsà la section Référence des AWS CLI commandes.

`cancel-update-stack`

L'exemple de code suivant montre comment utiliser`cancel-update-stack`.

AWS CLI

Pour annuler une mise à jour de la pile en cours

La `cancel-update-stack` commande suivante annule une mise à jour de la pile sur la `myteststack` pile :

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Pour plus de détails sur l'API, reportez-vous CancelUpdateStackà la section Référence des AWS CLI commandes.

`continue-update-rollback`

L'exemple de code suivant montre comment utiliser`continue-update-rollback`.

AWS CLI

Pour réessayer d'annuler une mise à jour

L'`continue-update-rollback`exemple suivant reprend une opération de restauration suite à un échec antérieur de la mise à jour de la pile.

```
aws cloudformation continue-update-rollback \
    --stack-name my-stack
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous ContinueUpdateRollbackà la section Référence des AWS CLI commandes.

`create-change-set`

L'exemple de code suivant montre comment utiliser`create-change-set`.

AWS CLI

Pour créer un ensemble de modifications

L'`create-change-set`exemple suivant crée un ensemble de modifications doté de `CAPABILITY_IAM` cette fonctionnalité. Le fichier `template.yaml` est un AWS CloudFormation modèle du dossier actuel qui définit une pile incluant des ressources IAM.

```
aws cloudformation create-change-set \
    --stack-name my-application \
    --change-set-name my-change-set \
    --template-body file://template.yaml \
    --capabilities CAPABILITY_IAM
```

Sortie :

```
{
    "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-application/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateChangeSetà la section Référence des AWS CLI commandes.

`create-stack-instances`

L'exemple de code suivant montre comment utiliser`create-stack-instances`.

AWS CLI

Pour créer des instances de pile

L'`create-stack-instances`exemple suivant crée des instances d'un ensemble de piles dans deux comptes et dans quatre régions. Le paramètre de tolérance aux pannes garantit que la mise à jour est tentée dans tous les comptes et régions, même si certaines piles ne peuvent pas être créées.

```
aws cloudformation create-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 223456789012 \
    --regions us-east-1 us-east-2 us-west-1 us-west-2 \
    --operation-preferences FailureToleranceCount=7
```

Sortie :

```
{
    "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"
}
```

Pour créer un ensemble de piles, utilisez la `create-stack-set` commande.

- Pour plus de détails sur l'API, reportez-vous CreateStackInstancesà la section Référence des AWS CLI commandes.

`create-stack-set`

L'exemple de code suivant montre comment utiliser`create-stack-set`.

AWS CLI

Pour créer un ensemble de piles

L'`create-stack-set`exemple suivant crée un ensemble de piles à l'aide du modèle de fichier YAML spécifié. `template.yaml`est un AWS CloudFormation modèle dans le dossier actuel qui définit une pile.

```
aws cloudformation create-stack-set \
    --stack-set-name my-stack-set \
    --template-body file://template.yaml \
    --description "SNS topic"
```

Sortie :

```
{
    "StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"
}
```

Pour ajouter des instances de pile à l'ensemble de piles, utilisez la `create-stack-instances` commande.

- Pour plus de détails sur l'API, reportez-vous CreateStackSetà la section Référence des AWS CLI commandes.

`create-stack`

L'exemple de code suivant montre comment utiliser`create-stack`.

AWS CLI

Pour créer une AWS CloudFormation pile

La `create-stacks` commande suivante crée une pile portant le nom `myteststack` à l'aide du `sampletemplate.json` modèle :

```
aws cloudformation create-stack --stack-name myteststack --template-body file://
sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
 ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Sortie :

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
```

```
}
```

Pour plus d'informations, consultez Stacks dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateStackà la section Référence des AWS CLI commandes.

`deactivate-type`

L'exemple de code suivant montre comment utiliser`deactivate-type`.

AWS CLI

Pour désactiver un type

L'`deactivate-type`exemple suivant désactive une extension publique précédemment activée dans ce compte et cette région.

```
aws cloudformation deactivate-type \
    --region us-west-2 \
    --type MODULE \
    --type-name Example::Test::Type::MODULE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeactivateTypeà la section Référence des AWS CLI commandes.

`delete-change-set`

L'exemple de code suivant montre comment utiliser`delete-change-set`.

AWS CLI

Pour supprimer un ensemble de modifications

L'`delete-change-set`exemple suivant supprime un ensemble de modifications en spécifiant le nom de l'ensemble de modifications et le nom de la pile.

```
aws cloudformation delete-change-set \
    --stack-name my-stack \
    --change-set-name my-change-set
```

Cette commande ne produit aucun résultat.

L'`delete-change-set`exemple suivant supprime un ensemble de modifications en spécifiant l'ARN complet du jeu de modifications.

```
aws cloudformation delete-change-set \
    --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteChangeSetà la section Référence des AWS CLI commandes.

`delete-stack-instances`

L'exemple de code suivant montre comment utiliser`delete-stack-instances`.

AWS CLI

Pour supprimer des instances de pile

L'`delete-stack-instances`exemple suivant supprime les instances d'une pile définie dans deux comptes dans deux régions et met fin aux piles.

```
aws cloudformation delete-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 567890123456 \
    --regions us-east-1 us-west-1 \
    --no-retain-stacks
```

Sortie :

```
{
    "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"
}
```

Pour supprimer un ensemble de piles vide, utilisez la `delete-stack-set` commande.

- Pour plus de détails sur l'API, reportez-vous DeleteStackInstancesà la section Référence des AWS CLI commandes.

`delete-stack-set`

L'exemple de code suivant montre comment utiliser`delete-stack-set`.

AWS CLI

Pour supprimer un ensemble de piles

La commande suivante supprime le jeu de piles vides spécifié. L'ensemble de piles doit être vide.

```
aws cloudformation delete-stack-set \
    --stack-set-name my-stack-set
```

Cette commande ne produit aucun résultat.

Pour supprimer des instances de l'ensemble de piles, utilisez la `delete-stack-instances` commande.

- Pour plus de détails sur l'API, reportez-vous DeleteStackSetà la section Référence des AWS CLI commandes.

`delete-stack`

L'exemple de code suivant montre comment utiliser`delete-stack`.

AWS CLI

Pour supprimer une pile

L'`delete-stack`exemple suivant supprime la pile spécifiée.

```
aws cloudformation delete-stack \
    --stack-name my-stack
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteStackà la section Référence des AWS CLI commandes.

`deploy`

L'exemple de code suivant montre comment utiliser`deploy`.

AWS CLI

La commande suivante déploie le modèle nommé `template.json` sur une pile nommée `my-new-stack` :

```
aws cloudformation deploy --template-file /path_to_template/template.json
 --stack-name my-new-stack --parameter-overrides Key1=Value1 Key2=Value2 --
tags Key1=Value1 Key2=Value2
```

- Pour plus de détails sur l'API, voir Déployer dans AWS CLI la référence des commandes.

`deregister-type`

L'exemple de code suivant montre comment utiliser`deregister-type`.

AWS CLI

Pour annuler l'enregistrement d'une version de type

L'`deregister-type`exemple suivant supprime la version de type spécifiée de l'utilisation active dans le CloudFormation registre, de sorte qu'elle ne puisse plus être utilisée dans les CloudFormation opérations.

```
aws cloudformation deregister-type \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --version-id 00000002
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du CloudFormation registre dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeregisterTypeà la section Référence des AWS CLI commandes.

`describe-account-limits`

L'exemple de code suivant montre comment utiliser`describe-account-limits`.

AWS CLI

Pour obtenir des informations sur les limites de votre compte

La commande suivante permet de récupérer une liste des limites régionales pour le compte courant.

```
aws cloudformation describe-account-limits
```

Sortie :

```
{
    "AccountLimits": [
        {
            "Name": "StackLimit",
            "Value": 200
        },
        {
            "Name": "StackOutputsLimit",
            "Value": 60
        },
        {
            "Name": "ConcurrentResourcesLimit",
            "Value": 2500
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAccountLimitsà la section Référence des AWS CLI commandes.

`describe-change-set`

L'exemple de code suivant montre comment utiliser`describe-change-set`.

AWS CLI

Pour obtenir des informations sur un ensemble de modifications

L'`describe-change-set`exemple suivant affiche les détails de l'ensemble de modifications spécifié par le nom de l'ensemble de modifications et le nom de la pile.

```
aws cloudformation describe-change-set \
    --change-set-name my-change-set \
    --stack-name my-stack
```

L'`describe-change-set`exemple suivant affiche les détails de l'ensemble de modifications spécifié par l'ARN complet du jeu de modifications :

```
aws cloudformation describe-change-set \
    --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

Sortie :

```
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "function",
                "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
                "ResourceType": "AWS::Lambda::Function",
                "Replacement": "False",
                "Scope": [
                    "Properties"
                ],
                "Details": [
                    {
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "Timeout",
                            "RequiresRecreation": "Never"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
```

```
                    }
                ]
            }
        }
    ],
    "ChangeSetName": "my-change-set",
    "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "StackName": "my-stack",
    "Description": null,
    "Parameters": null,
    "CreationTime": "2019-10-02T05:20:56.651Z",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {},
    "Capabilities": [
        "CAPABILITY_IAM"
    ],
    "Tags": null
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeChangeSetà la section Référence des AWS CLI commandes.

`describe-publisher`

L'exemple de code suivant montre comment utiliser`describe-publisher`.

AWS CLI

Pour décrire un éditeur

L'`describe-publisher`exemple suivant configure les informations pour un éditeur.

```
aws cloudformation describe-publisher \
    --region us-west-2 \
    --publisher-id 000q6TfUovXsEMmgKowxDZLlwqr2QUsh
```

Sortie :

```
{
    "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c",
    "PublisherStatus": "VERIFIED",
    "IdentityProvider": "AWS_Marketplace",
    "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?
id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribePublisherà la section Référence des AWS CLI commandes.

`describe-stack-drift-detection-status`

L'exemple de code suivant montre comment utiliser`describe-stack-drift-detection-status`.

AWS CLI

Pour vérifier l'état d'une opération de détection de dérive

L'`describe-stack-drift-detection-status`exemple suivant montre l'état d'une opération de détection de dérive. Obtenez le numéro d'identification lors de l'exécution de la `detect-stack-drift` commande.

```
aws cloudformation describe-stack-drift-detection-status \
    --stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

Sortie :

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4",
    "StackDriftStatus": "DRIFTED",
    "DetectionStatus": "DETECTION_COMPLETE",
    "DriftedStackResourceCount": 1,
    "Timestamp": "2019-10-02T05:54:30.902Z"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackDriftDetectionStatusà la section Référence des AWS CLI commandes.

`describe-stack-events`

L'exemple de code suivant montre comment utiliser`describe-stack-events`.

AWS CLI

Pour décrire les événements de la pile

L'`describe-stack-events`exemple suivant affiche les 2 événements les plus récents pour la pile spécifiée.

```
aws cloudformation describe-stack-events \
    --stack-name my-stack \
    --max-items 2

{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2019-10-02T05:34:29.556Z",
            "ResourceStatus": "UPDATE_COMPLETE"
        },
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2019-10-02T05:34:29.127Z",
            "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
```

```
        }
    ],

 "NextToken": "eyJOZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackEventsà la section Référence des AWS CLI commandes.

`describe-stack-instance`

L'exemple de code suivant montre comment utiliser`describe-stack-instance`.

AWS CLI

Pour décrire une instance de stack

La commande suivante décrit une instance de l'ensemble de piles spécifié dans le compte et la région spécifiés. L'ensemble de piles se trouve dans la région et le compte actuels, et l'instance se trouve dans la `us-west-2` région en compte`123456789012`. :

```
aws cloudformation describe-stack-instance \
    --stack-set-name my-stack-set \
    --stack-instance-account 123456789012 \
    --stack-instance-region us-west-2
```

Sortie :

```
{
    "StackInstance": {
        "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "Region": "us-west-2",
        "Account": "123456789012",
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/4287f9a0-e615-
xmpl-894a-12b31d3117be",
        "ParameterOverrides": [],
        "Status": "OUTDATED",
        "StatusReason": "ResourceLogicalId:ConfigBucket,
 ResourceType:AWS::S3::Bucket, ResourceStatusReason:You have attempted to create
 more buckets than allowed (Service: Amazon S3; Status Code: 400; Error Code:
```

```
 TooManyBuckets; Request ID: F7F21CXMPL580224; S3 Extended Request ID: egd/
Fdt89BXMPLyiqbMNljVk55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj967lhriaG/dWMobSO40o=)."
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackInstanceà la section Référence des AWS CLI commandes.

`describe-stack-resource-drifts`

L'exemple de code suivant montre comment utiliser`describe-stack-resource-drifts`.

AWS CLI

Pour obtenir des informations sur les ressources dérivées de la définition de la pile

La commande suivante affiche des informations sur les ressources dérivées pour la pile spécifiée. Pour lancer la détection de dérive, utilisez la `detect-stack-drift` commande. :

```
aws cloudformation describe-stack-resource-drifts \
    --stack-name my-stack
```

La sortie montre une fonction AWS Lambda qui a été modifiée : out-of-band

```
{
    "StackResourceDrifts": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":128,\"Role\":
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":900,\"TracingConfig\":{\"Mode\":\"Active\"}}",
            "ActualProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":256,\"Role\":
```

```
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":22,\"TracingConfig\":{\"Mode\":\"Active\"}}",
            "PropertyDifferences": [
                {
                    "PropertyPath": "/MemorySize",
                    "ExpectedValue": "128",
                    "ActualValue": "256",
                    "DifferenceType": "NOT_EQUAL"
                },
                {
                    "PropertyPath": "/Timeout",
                    "ExpectedValue": "900",
                    "ActualValue": "22",
                    "DifferenceType": "NOT_EQUAL"
                }
            ],
            "StackResourceDriftStatus": "MODIFIED",
            "Timestamp": "2019-10-02T05:54:44.064Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackResourceDriftsà la section Référence des AWS CLI commandes.

`describe-stack-resource`

L'exemple de code suivant montre comment utiliser`describe-stack-resource`.

AWS CLI

Pour obtenir des informations sur une ressource de pile

L'`describe-stack-resource`exemple suivant affiche les détails de la ressource nommée `MyFunction` dans la pile spécifiée.

```
aws cloudformation describe-stack-resource \
    --stack-name MyStack \
    --logical-resource-id MyFunction
```

Sortie :

```
{
    "StackResourceDetail": {
        "StackName": "MyStack",
        "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "LogicalResourceId": "MyFunction",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
        "ResourceStatus": "UPDATE_COMPLETE",
        "Metadata": "{}",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackResourceà la section Référence des AWS CLI commandes.

`describe-stack-resources`

L'exemple de code suivant montre comment utiliser`describe-stack-resources`.

AWS CLI

Pour obtenir des informations sur une ressource de pile

L'`describe-stack-resources`exemple suivant affiche les détails des ressources de la pile spécifiée.

```
aws cloudformation describe-stack-resources \
    --stack-name my-stack
```

Sortie :

```
{
    "StackResources": [
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
            "Timestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "Timestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "Timestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackResourcesà la section Référence des AWS CLI commandes.

`describe-stack-set-operation`

L'exemple de code suivant montre comment utiliser`describe-stack-set-operation`.

AWS CLI

Pour obtenir des informations sur une opération de stack set

L'exemple describe-stack-set-operation `suivant affiche les détails d'une opération de mise à jour sur le stack set spécifié.

```
aws cloudformation describe-stack-set-operation \
    --stack-set-name enable-config \
    --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

Sortie :

```
{
    "StackSetOperation": {
        "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",
        "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "Action": "UPDATE",
        "Status": "SUCCEEDED",
        "OperationPreferences": {
            "RegionOrder": [
                "us-east-1",
                "us-west-2",
                "eu-west-1",
                "us-west-1"
            ],
            "FailureToleranceCount": 7,
            "MaxConcurrentCount": 2
        },
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
        "CreationTimestamp": "2019-10-03T16:28:44.377Z",
        "EndTimestamp": "2019-10-03T16:42:08.607Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackSetOperationà la section Référence des AWS CLI commandes.

describe-stack-set

L'exemple de code suivant montre comment utiliserdescribe-stack-set.

AWS CLI

Pour obtenir des informations sur un ensemble de piles

L'exemple describe-stack-set `suivant affiche des détails sur l'ensemble de piles spécifié.

```
aws cloudformation describe-stack-set \
    --stack-set-name my-stack-set
```

Sortie :

```
{
    "StackSet": {
        "StackSetName": "my-stack-set",
        "StackSetId": "my-stack-set:296a3360-xmpl-40af-be78-9341e95bf743",
        "Description": "Create an Amazon SNS topic",
        "Status": "ACTIVE",
        "TemplateBody": "AWSTemplateFormatVersion: '2010-09-09'\nDescription: An AWS
 SNS topic\nResources:\n  topic:\n    Type: AWS::SNS::Topic",
        "Parameters": [],
        "Capabilities": [],
        "Tags": [],
        "StackSetARN": "arn:aws:cloudformation:us-west-2:123456789012:stackset/
enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStackSetà la section Référence des AWS CLI commandes.

describe-stacks

L'exemple de code suivant montre comment utiliserdescribe-stacks.

AWS CLI

Pour décrire les AWS CloudFormation piles

La `describe-stacks` commande suivante affiche des informations récapitulatives pour la `myteststack` pile :

```
aws cloudformation describe-stacks --stack-name myteststack
```

Sortie :

```
{
    "Stacks":  [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
            "Tags": [],
            "Outputs": [
                {
                    "Description": "Name of S3 bucket to hold website content",
                    "OutputKey": "BucketName",
                    "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
                }
            ],
            "StackStatusReason": null,
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false
        }
    ]
}
```

Pour plus d'informations, consultez Stacks dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeStacksà la section Référence des AWS CLI commandes.

`describe-type-registration`

L'exemple de code suivant montre comment utiliser`describe-type-registration`.

AWS CLI

Pour afficher les informations d'enregistrement du type

L'`describe-type-registration`exemple suivant affiche des informations sur l'enregistrement du type spécifié, notamment le statut, le type et la version actuels du type.

```
aws cloudformation describe-type-registration \
    --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "ProgressStatus": "COMPLETE",
    "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-
LogGroup",
    "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",
    "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-Logs-LogGroup/00000001"
}
```

Pour plus d'informations, consultez la section Utilisation du CloudFormation registre dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeTypeRegistrationà la section Référence des AWS CLI commandes.

`describe-type`

L'exemple de code suivant montre comment utiliser`describe-type`.

AWS CLI

Pour afficher les informations de type

L'`describe-type`exemple suivant affiche les informations relatives au type spécifié.

```
aws cloudformation describe-type \
```

```
    --type-name My::Logs::LogGroup \
    --type RESOURCE
```

Sortie :

```
{
    "SourceUrl": "https://github.com/aws-cloudformation/aws-cloudformation-resource-
providers-logs.git",
    "Description": "Customized resource derived from AWS::Logs::LogGroup",
    "TimeCreated": "2019-12-03T23:29:33.321Z",
    "Visibility": "PRIVATE",
    "TypeName": "My::Logs::LogGroup",
    "LastUpdated": "2019-12-03T23:29:33.321Z",
    "DeprecatedStatus": "LIVE",
    "ProvisioningType": "FULLY_MUTABLE",
    "Type": "RESOURCE",
    "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-
LogGroup/00000001",
    "Schema": "[details omitted]"
}
```

Pour plus d'informations, consultez la section Utilisation du CloudFormation registre dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeTypeà la section Référence des AWS CLI commandes.

`detect-stack-drift`

L'exemple de code suivant montre comment utiliser`detect-stack-drift`.

AWS CLI

Pour détecter les ressources dérivées

L'`detect-stack-drift`exemple suivant lance la détection de dérive pour la pile spécifiée.

```
aws cloudformation detect-stack-drift \
    --stack-name my-stack
```

Sortie :

```
{
    "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"
}
```

Vous pouvez ensuite utiliser cet identifiant avec la `describe-stack-resource-drifts` commande pour décrire les ressources dérivées.

- Pour plus de détails sur l'API, reportez-vous DetectStackDriftà la section Référence des AWS CLI commandes.

`detect-stack-resource-drift`

L'exemple de code suivant montre comment utiliser`detect-stack-resource-drift`.

AWS CLI

Pour détecter la dérive d'une ressource

L'`detect-stack-resource-drift`exemple suivant vérifie une ressource nommée `MyFunction` dans une pile nommée `MyStack` d'après drift :

```
aws cloudformation detect-stack-resource-drift \
   --stack-name MyStack \
   --logical-resource-id MyFunction
```

La sortie montre une fonction AWS Lambda qui a été modifiée : out-of-band

```
{
    "StackResourceDrift": {
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "LogicalResourceId": "MyFunction",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":128,\"Role\":
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":900,\"TracingConfig\":{\"Mode\":\"Active\"}}",
        "ActualProperties": "{\"Description\":\"Write a file to S3.\",\"Environment
\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf\"}},\"Handler\":
```

```
\"index.handler\",\"MemorySize\":256,\"Role\":\"arn:aws:iam::123456789012:role/
my-functionRole-HIZXMPLEOM9E\",\"Runtime\":\"nodejs10.x\",\"Tags\":[{\"Key\":
\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout\":22,\"TracingConfig\":{\"Mode\":
\"Active\"}}",
        "PropertyDifferences": [
            {
                "PropertyPath": "/MemorySize",
                "ExpectedValue": "128",
                "ActualValue": "256",
                "DifferenceType": "NOT_EQUAL"
            },
            {
                "PropertyPath": "/Timeout",
                "ExpectedValue": "900",
                "ActualValue": "22",
                "DifferenceType": "NOT_EQUAL"
            }
        ],
        "StackResourceDriftStatus": "MODIFIED",
        "Timestamp": "2019-10-02T05:58:47.433Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DetectStackResourceDriftà la section Référence des AWS CLI commandes.

`detect-stack-set-drift`

L'exemple de code suivant montre comment utiliser`detect-stack-set-drift`.

AWS CLI

Pour détecter la dérive sur un ensemble de piles et toutes les instances de pile associées

L'`detect-stack-set-drift`exemple suivant lance des opérations de détection de dérive sur l'ensemble de piles spécifié, y compris toutes les instances de pile associées à cet ensemble de piles, et renvoie un ID d'opération qui peut être utilisé pour suivre l'état de l'opération de dérive.

```
aws cloudformation detect-stack-set-drift \
    --stack-set-name stack-set-drift-example
```

Sortie :

```
{
    "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Détection des modifications de configuration non gérées dans les Stack Sets dans le Guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous DetectStackSetDriftà la section Référence des AWS CLI commandes.

`estimate-template-cost`

L'exemple de code suivant montre comment utiliser`estimate-template-cost`.

AWS CLI

Pour estimer le coût du modèle

L'`estimate-template-cost`exemple suivant génère une estimation des coûts pour un modèle nommé `template.yaml` dans le dossier actuel.

```
aws cloudformation estimate-template-cost \
    --template-body file://template.yaml
```

Sortie :

```
{
    "Url": "http://calculator.s3.amazonaws.com/calc5.html?
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"
}
```

- Pour plus de détails sur l'API, reportez-vous EstimateTemplateCostà la section Référence des AWS CLI commandes.

`execute-change-set`

L'exemple de code suivant montre comment utiliser`execute-change-set`.

AWS CLI

Pour exécuter un ensemble de modifications

L'`execute-change-set`exemple suivant exécute un ensemble de modifications spécifié par le nom de l'ensemble de modifications et le nom de la pile.

```
aws cloudformation execute-change-set \
    --change-set-name my-change-set \
    --stack-name my-stack
```

L'`execute-change-set`exemple suivant exécute un ensemble de modifications spécifié par l'ARN complet du jeu de modifications.

```
aws cloudformation execute-change-set \
    --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- Pour plus de détails sur l'API, reportez-vous ExecuteChangeSetà la section Référence des AWS CLI commandes.

`get-stack-policy`

L'exemple de code suivant montre comment utiliser`get-stack-policy`.

AWS CLI

Pour consulter une politique de stack

L'`get-stack-policy`exemple suivant affiche la politique de pile pour la pile spécifiée. Pour associer une politique à une pile, utilisez la `set-stack-policy` commande.

```
aws cloudformation get-stack-policy \
    --stack-name my-stack
```

Sortie :

```
{
    "StackPolicyBody": "{\n  \"Statement\" : [\n    {\n      \"Effect\" :
 \"Allow\",\n      \"Action\" : \"Update:*\",\n      \"Principal\": \"*\",\n
     \"Resource\" : \"*\"\n    },\n    {\n      \"Effect\" : \"Deny\",\n
 \"Action\" : \"Update:*\",\n      \"Principal\": \"*\",\n      \"Resource\" :
 \"LogicalResourceId/bucket\"\n    }\n  ]\n}\n"
}
```

- Pour plus de détails sur l'API, reportez-vous GetStackPolicyà la section Référence des AWS CLI commandes.

`get-template-summary`

L'exemple de code suivant montre comment utiliser`get-template-summary`.

AWS CLI

Pour afficher le résumé d'un modèle

La commande suivante affiche des informations récapitulatives sur les ressources et les métadonnées du fichier modèle spécifié.

```
aws cloudformation get-template-summary \
   --template-body file://template.yaml
```

Sortie :

```
{
    "Parameters": [],
    "Description": "A VPC and subnets.",
    "ResourceTypes": [
        "AWS::EC2::VPC",
        "AWS::EC2::Subnet",
        "AWS::EC2::Subnet",
        "AWS::EC2::RouteTable",
        "AWS::EC2::VPCEndpoint",
        "AWS::EC2::SubnetRouteTableAssociation",
        "AWS::EC2::SubnetRouteTableAssociation",
        "AWS::EC2::VPCEndpoint"
    ],
    "Version": "2010-09-09"
}
```

- Pour plus de détails sur l'API, reportez-vous GetTemplateSummaryà la section Référence des AWS CLI commandes.

`get-template`

L'exemple de code suivant montre comment utiliser`get-template`.

AWS CLI

Pour afficher le corps du modèle d'une AWS CloudFormation pile

La `get-template` commande suivante montre le modèle de la `myteststack` pile :

```
aws cloudformation get-template --stack-name myteststack
```

Sortie :

```
{
    "TemplateBody": {
        "AWSTemplateFormatVersion": "2010-09-09",
        "Outputs": {
            "BucketName": {
                "Description": "Name of S3 bucket to hold website content",
                "Value": {
                    "Ref": "S3Bucket"
                }
            }
        },
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
        "Resources": {
            "S3Bucket": {
                "Type": "AWS::S3::Bucket",
                "Properties": {
                    "AccessControl": "PublicRead"
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetTemplateà la section Référence des AWS CLI commandes.

`list-change-sets`

L'exemple de code suivant montre comment utiliser`list-change-sets`.

AWS CLI

Pour répertorier les ensembles de modifications

L'`list-change-sets`exemple suivant affiche une liste des ensembles de modifications en attente pour la pile spécifiée.

```
aws cloudformation list-change-sets \
   --stack-name my-stack
```

Sortie :

```
{
    "Summaries": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "StackName": "my-stack",
            "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/
my-change-set/70160340-7914-xmpl-bcbf-128a1fa78b5d",
            "ChangeSetName": "my-change-set",
            "ExecutionStatus": "AVAILABLE",
            "Status": "CREATE_COMPLETE",
            "CreationTime": "2019-10-02T05:38:54.297Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListChangeSetsà la section Référence des AWS CLI commandes.

`list-exports`

L'exemple de code suivant montre comment utiliser`list-exports`.

AWS CLI

Pour répertorier les exportations

L'`list-exports`exemple suivant affiche une liste des exportations à partir des piles de la région actuelle.

```
aws cloudformation list-exports
```

Sortie :

```
{
    "Exports": [
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-subnet-a",
            "Value": "subnet-07b410xmplddcfa03"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-subnet-b",
            "Value": "subnet-075ed3xmplebd2fb1"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-vpcid",
            "Value": "vpc-011d7xmpl100e9841"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListExportsà la section Référence des AWS CLI commandes.

`list-imports`

L'exemple de code suivant montre comment utiliser`list-imports`.

AWS CLI

Pour répertorier les importations

L'`list-imports`exemple suivant répertorie les piles qui importent l'exportation spécifiée. Pour obtenir la liste des exportations disponibles, utilisez la `list-exports` commande.

```
aws cloudformation list-imports \
```

```
    --export-name private-vpc-vpcid
```

Sortie :

```
{
    "Imports": [
        "my-database-stack"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListImportsà la section Référence des AWS CLI commandes.

`list-stack-instances`

L'exemple de code suivant montre comment utiliser`list-stack-instances`.

AWS CLI

Pour répertorier les instances d'une pile

L'`list-stack-instances`exemple suivant répertorie les instances créées à partir de l'ensemble de piles spécifié.

```
aws cloudformation list-stack-instances \
    --stack-set-name enable-config
```

L'exemple de sortie inclut des détails sur une pile qui n'a pas pu être mise à jour en raison d'une erreur :

```
{
    "Summaries": [
        {
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Region": "us-west-2",
            "Account": "123456789012",
            "StackId": "arn:aws:cloudformation:ap-northeast-1:123456789012:stack/
StackSet-enable-config-35a6ac50-d9f8-4084-86e4-7da34d5de4c4/a1631cd0-e5fb-xmpl-
b474-0aa20f14f06e",
            "Status": "CURRENT"
        },
        {
```

```
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Region": "us-west-2",
            "Account": "123456789012",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/eab53680-e5fa-xmpl-
ba14-0a522351f81e",
            "Status": "OUTDATED",
            "StatusReason": "ResourceLogicalId:ConfigDeliveryChannel,
 ResourceType:AWS::Config::DeliveryChannel, ResourceStatusReason:Failed to put
 delivery channel 'StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532-
ConfigDeliveryChannel-1OJWJ7XD59WR0' because the maximum number of delivery
 channels: 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code:
 MaxNumberOfDeliveryChannelsExceededException; Request ID: d14b34a0-ef7c-xmpl-
acf8-8a864370ae56)."
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListStackInstancesà la section Référence des AWS CLI commandes.

`list-stack-resources`

L'exemple de code suivant montre comment utiliser`list-stack-resources`.

AWS CLI

Pour répertorier les ressources d'une pile

La commande suivante affiche la liste des ressources de la pile spécifiée.

```
aws cloudformation list-stack-resources \
    --stack-name my-stack
```

Sortie :

```
{
    "StackResourceSummaries": [
        {
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
```

```
            "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListStackResourcesà la section Référence des AWS CLI commandes.

`list-stack-set-operation-results`

L'exemple de code suivant montre comment utiliser`list-stack-set-operation-results`.

AWS CLI

Pour répertorier les résultats des opérations d'un ensemble de piles

La commande suivante affiche les résultats d'une opération de mise à jour sur les instances du stack set spécifié.

```
aws cloudformation list-stack-set-operation-results \
    --stack-set-name enable-config \
    --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

Sortie :

```
{
    "Summaries": [
        {
            "Account": "223456789012",
            "Region": "us-west-2",
            "Status": "SUCCEEDED",
            "AccountGateResult": {
                "Status": "SKIPPED",
                "StatusReason": "Function not found: arn:aws:lambda:eu-
west-1:223456789012:function:AWSCloudFormationStackSetAccountGate"
            }
        },
        {
            "Account": "223456789012",
            "Region": "ap-south-1",
            "Status": "CANCELLED",
            "StatusReason": "Cancelled since failure tolerance has exceeded"
        }
    ]
}
```

Remarque : Le `SKIPPED` statut de `AccountGateResult` est attendu pour les opérations réussies, sauf si vous créez une fonction de portail de compte.

- Pour plus de détails sur l'API, reportez-vous ListStackSetOperationResultsà la section Référence des AWS CLI commandes.

`list-stack-set-operations`

L'exemple de code suivant montre comment utiliser`list-stack-set-operations`.

AWS CLI

Pour répertorier les opérations relatives aux ensembles de piles

L'`list-stack-set-operations`exemple suivant affiche la liste des opérations les plus récentes sur l'ensemble de piles spécifié.

```
aws cloudformation list-stack-set-operations \
    --stack-set-name my-stack-set
```

Sortie :

```
{
    "Summaries": [
        {
            "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",
            "Action": "UPDATE",
            "Status": "SUCCEEDED",
            "CreationTimestamp": "2019-10-03T16:28:44.377Z",
            "EndTimestamp": "2019-10-03T16:42:08.607Z"
        },
        {
            "OperationId": "891aa98f-7118-xmpl-00b2-00954d1dd0d6",
            "Action": "UPDATE",
            "Status": "FAILED",
            "CreationTimestamp": "2019-10-03T15:43:53.916Z",
            "EndTimestamp": "2019-10-03T15:45:58.925Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListStackSetOperationsà la section Référence des AWS CLI commandes.

`list-stack-sets`

L'exemple de code suivant montre comment utiliser`list-stack-sets`.

AWS CLI

Pour répertorier les ensembles de piles

L'`list-stack-sets`exemple suivant affiche la liste des ensembles de piles dans la région et le compte actuels.

```
aws cloudformation list-stack-sets
```

Sortie :

```
{
    "Summaries": [
        {
            "StackSetName": "enable-config",
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Description": "Enable AWS Config",
            "Status": "ACTIVE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListStackSetsà la section Référence des AWS CLI commandes.

`list-stacks`

L'exemple de code suivant montre comment utiliser`list-stacks`.

AWS CLI

Pour répertorier les AWS CloudFormation piles

La `list-stacks` commande suivante affiche un résumé de toutes les piles dont le statut est : `CREATE_COMPLETE`

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Sortie :

```
[
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
        "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
```

```
        "StackStatusReason": null,
        "CreationTime": "2013-08-26T03:27:10.190Z",
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE"
    }
]
```

- Pour plus de détails sur l'API, reportez-vous ListStacksà la section Référence des AWS CLI commandes.

`list-type-registrations`

L'exemple de code suivant montre comment utiliser`list-type-registrations`.

AWS CLI

Pour répertorier les enregistrements terminés d'un type

L'`list-type-registrations`exemple suivant affiche une liste des enregistrements de type terminés pour le type spécifié.

```
aws cloudformation list-type-registrations \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --registration-status-filter COMPLETE
```

Sortie :

```
{
    "RegistrationTokenList": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du CloudFormation registre dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTypeRegistrationsà la section Référence des AWS CLI commandes.

`list-type-versions`

L'exemple de code suivant montre comment utiliser`list-type-versions`.

AWS CLI

Pour répertorier la version d'une extension

L'`list-type-versions`exemple suivant renvoie des informations récapitulatives sur les versions d'une extension.

```
aws cloudformation list-type-versions \
  --endpoint https://example.com \
  --region us-west-2 \
  --type RESOURCE \
  --type-name My::Resource::Example \
  --publisher-id 123456789012
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTypeVersionsà la section Référence des AWS CLI commandes.

`list-types`

L'exemple de code suivant montre comment utiliser`list-types`.

AWS CLI

Pour répertorier les types de ressources privées d'un compte

L'`list-types`exemple suivant affiche une liste des types de ressources privées actuellement enregistrés dans le AWS compte courant.

```
aws cloudformation list-types
```

Sortie :

```
{
```

```
    "TypeSummaries": [
        {
            "Description": "WordPress blog resource for internal use",
            "LastUpdated": "2019-12-04T18:28:15.059Z",
            "TypeName": "My::WordPress::BlogExample",
            "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-WordPress-BlogExample",
            "DefaultVersionId": "00000005",
            "Type": "RESOURCE"
        },
        {
            "Description": "Customized resource derived from AWS::Logs::LogGroup",
            "LastUpdated": "2019-12-04T18:28:15.059Z",
            "TypeName": "My::Logs::LogGroup",
            "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-Logs-LogGroup",
            "DefaultVersionId": "00000003",
            "Type": "RESOURCE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du CloudFormation registre dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTypesà la section Référence des AWS CLI commandes.

`package`

L'exemple de code suivant montre comment utiliser`package`.

AWS CLI

La commande suivante exporte un modèle nommé `template.json` en téléchargeant des artefacts locaux dans le compartiment S3 `bucket-name` et écrit le modèle exporté dans : `packaged-template.json`

```
aws cloudformation package --template-file /path_to_template/template.json --s3-
bucket bucket-name --output-template-file packaged-template.json --use-json
```

- Pour plus de détails sur l'API, voir Package dans AWS CLI Command Reference.

`publish-type`

L'exemple de code suivant montre comment utiliser`publish-type`.

AWS CLI

Pour publier une extension

L'`publish-type`exemple suivant publie l'extension spécifiée dans le CloudFormation registre en tant qu'extension publique dans cette région.

```
aws cloudformation publish-type \
  --region us-west-2 \
  --type RESOURCE \
  --type-name Example::Test::1234567890abcdef0
```

Sortie :

```
{
    "PublicTypeArn":"arn:aws:cloudformation:us-west-2::type/
resource/000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c/Example-
Test-1234567890abcdef0/1.0.0"
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous PublishTypeà la section Référence des AWS CLI commandes.

`register-publisher`

L'exemple de code suivant montre comment utiliser`register-publisher`.

AWS CLI

Pour enregistrer un éditeur

L'`register-publisher`exemple suivant enregistre un éditeur et accepte le paramètre terms and condition.

```
aws cloudformation register-publisher \
```

```
  --region us-west-2 \
  --accept-terms-and-conditions
```

Sortie :

```
{
    "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c"
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous RegisterPublisherà la section Référence des AWS CLI commandes.

`register-type`

L'exemple de code suivant montre comment utiliser`register-type`.

AWS CLI

Pour enregistrer un type de ressource

L'`register-type`exemple suivant enregistre le type de ressource spécifié en tant que type de ressource privée dans le compte de l'utilisateur.

```
aws cloudformation register-type \
    --type-name My::Organization::ResourceName \
    --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \
    --type RESOURCE
```

Sortie :

```
{
    "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"
}
```

Pour plus d'informations, consultez la section Enregistrement des fournisseurs de ressources dans le guide de l'utilisateur de l'interface de ligne de CloudFormation commande pour le développement de types.

- Pour plus de détails sur l'API, reportez-vous RegisterTypeà la section Référence des AWS CLI commandes.

`set-stack-policy`

L'exemple de code suivant montre comment utiliser`set-stack-policy`.

AWS CLI

Pour appliquer une politique de stack

L'`set-stack-policy`exemple suivant désactive les mises à jour pour la ressource spécifiée dans la pile spécifiée. `stack-policy.json`est un document JSON qui définit les opérations autorisées sur les ressources de la pile.

```
aws cloudformation set-stack-policy \
    --stack-name my-stack \
    --stack-policy-body file://stack-policy.json
```

Sortie :

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/bucket"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous SetStackPolicyà la section Référence des AWS CLI commandes.

`set-type-configuration`

L'exemple de code suivant montre comment utiliser`set-type-configuration`.

AWS CLI

Pour configurer les données

L'`set-type-configuration`exemple suivant indique les données de configuration pour une CloudFormation extension enregistrée, dans le compte et la région donnés.

```
aws cloudformation set-type-configuration \
    --region us-west-2 \
    --type RESOURCE \
    --type-name Example::Test::Type \
    --configuration-alias default \
    --configuration "{\"CredentialKey\": \"testUserCredential\"}"
```

Sortie :

```
{
    "ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-
configuration/resource/Example-Test-Type/default"
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous SetTypeConfigurationà la section Référence des AWS CLI commandes.

`set-type-default-version`

L'exemple de code suivant montre comment utiliser`set-type-default-version`.

AWS CLI

Pour définir la version par défaut d'un type

L'`set-type-default-version`exemple suivant définit la version du type spécifié à utiliser par défaut pour ce type.

```
aws cloudformation set-type-default-version \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --version-id 00000003
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du CloudFormation registre dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous SetTypeDefaultVersionà la section Référence des AWS CLI commandes.

`signal-resource`

L'exemple de code suivant montre comment utilisersignal-resource.

AWS CLI

Pour signaler une ressource

L'signal-resourceexemple suivant indique que `success` la condition d'attente spécifiée `MyWaitCondition` dans la pile nommée doit être rempliemy-stack.

```
aws cloudformation signal-resource \
    --stack-name my-stack \
    --logical-resource-id MyWaitCondition \
    --unique-id 1234 \
    --status SUCCESS
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SignalResourceà la section Référence des AWS CLI commandes.

`stop-stack-set-operation`

L'exemple de code suivant montre comment utiliserstop-stack-set-operation.

AWS CLI

Pour arrêter une opération de stack set

L'`stop-stack-set-operation`exemple suivant arrête une opération de mise à jour en cours sur le stack set spécifié.

```
aws cloudformation stop-stack-set-operation \
    --stack-set-name my-stack-set \
    --operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous StopStackSetOperationà la section Référence des AWS CLI commandes.

`test-type`

L'exemple de code suivant montre comment utiliser`test-type`.

AWS CLI

Pour tester une extension

L'`test-type`exemple suivant teste une extension enregistrée pour s'assurer qu'elle répond à toutes les exigences nécessaires pour être publiée dans le CloudFormation registre.

```
aws cloudformation test-type \
    --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001
```

Sortie :

```
{
    "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001"
}
```

Pour plus d'informations, consultez la section Utilisation du AWS CloudFormation registre dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous TestTypeà la section Référence des AWS CLI commandes.

`update-stack-instances`

L'exemple de code suivant montre comment utiliser`update-stack-instances`.

AWS CLI

Pour mettre à jour les instances de stack

L'`update-stack-instances`exemple suivant tente à nouveau de mettre à jour les instances de stack de deux comptes situés dans deux régions avec les paramètres les plus récents. Le paramètre de tolérance aux pannes spécifié garantit que la mise à jour est tentée dans tous les comptes et régions, même si certaines piles ne peuvent pas être mises à jour.

```
aws cloudformation update-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 567890123456 \
    --regions us-east-1 us-west-2 \
    --operation-preferences FailureToleranceCount=3
```

Sortie :

```
{
    "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateStackInstancesà la section Référence des AWS CLI commandes.

`update-stack-set`

L'exemple de code suivant montre comment utiliser`update-stack-set`.

AWS CLI

Pour mettre à jour un ensemble de piles

L'`update-stack-set`exemple suivant ajoute une balise avec le nom de la clé `Owner` et une valeur de `IT` aux instances de pile de l'ensemble de piles spécifié.

```
aws cloudformation update-stack-set \
    --stack-set-name my-stack-set \
    --use-previous-template \
```

```
    --tags Key=Owner,Value=IT
```

Sortie :

```
{
    "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateStackSetà la section Référence des AWS CLI commandes.

`update-stack`

L'exemple de code suivant montre comment utiliser`update-stack`.

AWS CLI

Pour mettre à jour les AWS CloudFormation piles

La `update-stack` commande suivante met à jour le modèle et les paramètres d'entrée de la `mystack` pile :

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template --
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair
 ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

La `update-stack` commande suivante met à jour uniquement la valeur du `SubnetIDs` paramètre de la `mystack` pile. Si vous ne spécifiez aucune valeur de paramètre, la valeur par défaut spécifiée dans le modèle est utilisée :

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template
 --parameters ParameterKey=KeyPairName,UsePreviousValue=true
 ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

La `update-stack` commande suivante ajoute deux sujets de notification de pile à la `mystack` pile :

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --
notification-arns "arn:aws:sns:use-east-1:123456789012:mytopic1" "arn:aws:sns:us-
east-1:123456789012:mytopic2"
```

Pour plus d'informations, consultez la section AWS CloudFormation Stack Updates dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateStackà la section Référence des AWS CLI commandes.

`update-termination-protection`

L'exemple de code suivant montre comment utiliser`update-termination-protection`.

AWS CLI

Pour activer la protection contre le licenciement

L'`update-termination-protection`exemple suivant active la protection des terminaisons sur la pile spécifiée.

```
aws cloudformation update-termination-protection \
    --stack-name my-stack \
    --enable-termination-protection
```

Sortie :

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateTerminationProtectionà la section Référence des AWS CLI commandes.

`validate-template`

L'exemple de code suivant montre comment utiliser`validate-template`.

AWS CLI

Pour valider un AWS CloudFormation modèle

La `validate-template` commande suivante valide le `sampletemplate.json` modèle :

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Sortie :

```
{
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
 showing how to create a publicly accessible S3 bucket. **WARNING** This template
 creates an S3 bucket. You will be billed for the AWS resources used if you create a
 stack from this template.",
    "Parameters": [],
    "Capabilities": []
}
```

Pour plus d'informations, consultez la section Utilisation des AWS CloudFormation modèles dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ValidateTemplateà la section Référence des AWS CLI commandes.

CloudFront exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CloudFront.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-distribution-tenant-web-acl`

L'exemple de code suivant montre comment utiliser`associate-distribution-tenant-web-acl`.

AWS CLI

Pour associer une ACL Web à un locataire CloudFront de distribution

L'`associate-distribution-tenant-web-acl`exemple suivant associe une ACL Web à une CloudFront distribution avec ETag `E13V1IB3VIYABC`.

```
aws cloudfront associate-distribution-tenant-web-acl \
    --id dt_2wjDZi3hD1ivOXf6rpZJO1AB \
    --if-match E13V1IB3VIYABC \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f37123ABC
```

Sortie :

```
{
    "ETag": "E1VC38T7YXBABC",
    "Id": "dt_2wjDZi3hD1ivOXf6rpZJO1AB",
    "WebACLArn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f37123ABC"
}
```

Pour plus d'informations, consultez la section Utiliser les protections AWS WAF dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateDistributionTenantWebAclà la section Référence des AWS CLI commandes.

`associate-distribution-web-acl`

L'exemple de code suivant montre comment utiliser`associate-distribution-web-acl`.

AWS CLI

Pour associer une ACL Web à une CloudFront distribution

L'`associate-distribution-web-acl`exemple suivant associe une ACL Web à une CloudFront distribution.

```
aws cloudfront associate-distribution-web-acl \
    --id E1XNX8R2GOAABC \
    --if-match E2YWS1C2J30ABC \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f3746cABC
```

Sortie :

```
{
    "ETag": "E3QE7ED60U0ABC",
    "Id": "E1XNX8R2GOAABC",
    "WebACLArn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f3746cABC"
}
```

Pour plus d'informations, consultez la section Utiliser les protections AWS WAF dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateDistributionWebAclà la section Référence des AWS CLI commandes.

`create-cloud-front-origin-access-identity`

L'exemple de code suivant montre comment utiliser`create-cloud-front-origin-access-identity`.

AWS CLI

Pour créer une identité CloudFront d'accès à l'origine

L'exemple suivant crée une identité CloudFront d'accès à l'origine (OAI) en fournissant la configuration OAI sous forme d'argument de ligne de commande :

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config \
        CallerReference="cli-example",Comment="Example OAI"
```

Vous pouvez accomplir la même chose en fournissant la configuration OAI dans un fichier JSON, comme illustré dans l'exemple suivant :

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config file://OAI-config.json
```

Le fichier `OAI-config.json` est un document JSON situé dans le répertoire actuel qui contient les éléments suivants :

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI"
}
```

Que vous fournissiez la configuration OAI avec un argument de ligne de commande ou un fichier JSON, le résultat est le même :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/origin-access-identity/
cloudfront/E74FTE3AEXAMPLE",
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateCloudFrontOriginAccessIdentityà la section Référence des AWS CLI commandes.

`create-connection-group`

L'exemple de code suivant montre comment utiliser`create-connection-group`.

AWS CLI

Pour créer un groupe de connexion dans CloudFront

L'`create-connection-group`exemple suivant crée un groupe de connexion activé, spécifie une liste d'adresses IP statiques Anycast et le IPv6 désactive.

```
aws cloudfront create-connection-group \
    --name cg-with-anycast-ip-list \
    --no-ipv6-enabled \
    --enabled \
    --anycast-ip-list-id aip_CCkW6gKrDiBD4n78123ABC \
    --tags "Items=[{Key=abc,Value=123}]"
```

Sortie :

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2yb6uj74B4PCbfhT31WFdiSABC",
        "Name": "cg-with-anycast-ip-list",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2yb6uj74B4PCbfhT31WFdiSABC",
        "CreatedTime": "2025-06-16T16:25:50.061000+00:00",
        "LastModifiedTime": "2025-06-16T16:25:50.061000+00:00",
        "Tags": {
            "Items": [
                {
                    "Key": "abc",
                    "Value": "123"
                }
            ]
        },
        "Ipv6Enabled": false,
        "RoutingEndpoint": "dj6xusxq65abc.cloudfront.net",
        "AnycastIpListId": "aip_CCkW6gKrDiBD4n78123ABC",
        "Status": "InProgress",
        "Enabled": true,
        "IsDefault": false
    }
}
```

Pour plus d'informations, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateConnectionGroupà la section Référence des AWS CLI commandes.

`create-distribution-tenant`

L'exemple de code suivant montre comment utiliser`create-distribution-tenant`.

AWS CLI

Exemple 1 : pour créer un tenant de CloudFront distribution utilisant un certificat personnalisé

L'`create-distribution-tenant`exemple suivant crée un tenant de CloudFront distribution qui spécifie des personnalisations pour désactiver le WAF, ajouter des restrictions géographiques et utiliser un autre certificat TLS.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

Contenu de `tenant.json` :

```
{
    "DistributionId": "E1XNX8R2GOAABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "testParam",
            "Value": "defaultValue"
        }
    ],
    "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
    "Enabled": false,
    "Tags": {
        "Items": [
            {
                "Key": "tag",
```

```
                "Value": "tagValue"
            }
        ]
    },
    "Name": "new-tenant-customizations",
    "Customizations": {
        "GeoRestrictions": {
            "Locations": ["DE"],
            "RestrictionType": "whitelist"
        },
        "WebAcl": {
            "Action": "disable"
        },
        "Certificate": {
            "Arn": "arn:aws:acm:us-east-1:123456789012:certificate/ec53f564-
ea5a-4e4a-a0a2-e3c989449abc"
        }
    }
}
```

Sortie :

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2yN5tYwVbPKr7m2IB69M1yp1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "new-tenant-customizations",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yN5tYwVbPKr7m2IB69M1yp1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Tags": {
            "Items": [
                {
                    "Key": "tag",
                    "Value": "tagValue"
                }
            ]
```

```
        },
        "Customizations": {
            "WebAcl": {
                "Action": "disable"
            },
            "Certificate": {
                "Arn": "arn:aws:acm:us-east-1:123456789012:certificate/ec53f564-
ea5a-4e4a-a0a2-e3c989449abc"
            },
            "GeoRestrictions": {
                "RestrictionType": "whitelist",
                "Locations": [
                    "DE"
                ]
            }
        },
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "defaultValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-06-11T17:20:06.432000+00:00",
        "LastModifiedTime": "2025-06-11T17:20:06.432000+00:00",
        "Enabled": false,
        "Status": "InProgress"
    }
}
```

Exemple 2 : pour créer un tenant de distribution avec un certificat hérité

L'`create-distribution-tenant`exemple suivant crée un tenant de distribution et spécifie un certificat TLS hérité de la distribution multi-locataires.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

Contenu de `tenant.json` :

```
{
    "DistributionId": "E1HVIAU7U12ABC",
    "Domains": [
```

```
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "Enabled": true,
    "Name": "new-tenant-no-cert"
}
```

Sortie :

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhRB0vBe0B72LZCVy1mgzI1AB",
        "DistributionId": "E1HVIAU7U12ABC",
        "Name": "new-tenant-no-cert",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhRB0vBe0B72LZCVy1mgzI1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
        "CreatedTime": "2025-07-10T20:59:38.414000+00:00",
        "LastModifiedTime": "2025-07-10T20:59:38.414000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

Exemple 3 : pour créer un tenant de CloudFront distribution à l'aide d'un jeton de validation CloudFront hébergé par -hosted

L'`create-distribution-tenant`exemple suivant crée un tenant de distribution et utilise un jeton de validation CloudFront -hosted pour votre nom de domaine.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

Contenu de `tenant.json` :

```
{
    "DistributionId": "E2GJ5J9QN12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
    "Enabled": true,
    "Name": "new-tenant-cf-hosted",
    "ManagedCertificateRequest": {
        "ValidationTokenHost": "cloudfront"
    }
}
```

Important : pour exécuter correctement cette commande, vous devez configurer un enregistrement DNS CNAME qui pointe votre nouveau domaine (exemple.com) vers le point de terminaison de routage du groupe de connexion associé au locataire de distribution. Cet enregistrement CNAME doit également être propagé pour que cette demande CloudFront puisse être traitée avec succès.

Sortie :

```
{
    "ETag": "E23ZP02F0ABC",
```

```
    "DistributionTenant": {
        "Id": "dt_2zhStKrA524GvvTWJX92Ozl1AB",
        "DistributionId": "E2GJ5J9QN12ABC",
        "Name": "new-tenant-cf-hosted",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhStKrA524GvvTWJX92Ozl1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2zhSaGatwwXjTjE42nneZzqABC",
        "CreatedTime": "2025-07-10T21:13:46.416000+00:00",
        "LastModifiedTime": "2025-07-10T21:13:46.416000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

Exemple 4 : pour créer un tenant de CloudFront distribution à l'aide d'un jeton de validation auto-hébergé

L'`create-distribution-tenant`exemple suivant crée un tenant CloudFront de distribution et utilise un jeton de validation auto-hébergé.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

Contenu de `tenant.json` :

```
{
    "DistributionId": "E2GJ5J9QN12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
```

```
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "Enabled": true,
    "Name": "new-tenant-self-hosted",
    "ManagedCertificateRequest": {
        "ValidationTokenHost": "self-hosted"
    }
}
```

Sortie :

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhTFBV93OfFJJ3YMdNM5BC1AB",
        "DistributionId": "E2GJ5J9QN12ABC",
        "Name": "new-tenant-self-hosted",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhTFBV93OfFJJ3YMdNM5BC1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
        "CreatedTime": "2025-07-10T21:16:39.828000+00:00",
        "LastModifiedTime": "2025-07-10T21:16:39.828000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

Important : Après avoir exécuté cette commande, le tenant de distribution sera créé sans validation. Pour valider la demande de certificat géré et configurer le DNS lorsque vous êtes prêt à commencer à recevoir du trafic, consultez la section Configuration complète du domaine dans le manuel Amazon CloudFront Developer Guide.

Pour plus d'informations sur la création de locataires de distribution, consultez la section Créer une distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDistributionTenantà la section Référence des AWS CLI commandes.

`create-distribution-with-tags`

L'exemple de code suivant montre comment utiliser`create-distribution-with-tags`.

AWS CLI

Pour créer une CloudFront distribution avec des balises

L'`create-distribution-with-tags`exemple suivant crée une distribution avec deux balises en fournissant la configuration de distribution et les balises dans un fichier JSON nommé`dist-config-with-tags.json`.

```
aws cloudfront create-distribution-with-tags \
    --distribution-config-with-tags file://dist-config-with-tags.json
```

Le fichier `dist-config-with-tags.json` est un document JSON situé dans le dossier actuel. Notez l'`Tags`objet en haut du fichier, qui contient deux balises :

`Name = ExampleDistributionProject = ExampleProject`

Contenu de `dist-config-with-tags.json` :

```
{
    "Tags": {
        "Items": [
            {
                "Key": "Name",
                "Value": "ExampleDistribution"
            },
            {
                "Key": "Project",
```

```
                "Value": "ExampleProject"
            }
        ]
    },
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
```

```
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
```

```
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE",
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-04T23:35:41.433Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
```

```
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
```

```
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
```

```
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDistributionWithTagsà la section Référence des AWS CLI commandes.

`create-distribution`

L'exemple de code suivant montre comment utiliser`create-distribution`.

AWS CLI

Exemple 1 : pour créer une CloudFront distribution

L'`create-distribution`exemple suivant crée une distribution pour un compartiment S3 nomméamzn-s3-demo-bucket, et le spécifie également `index.html` comme objet racine par défaut, à l'aide d'arguments de ligne de commande.

```
aws cloudfront create-distribution \
    --origin-domain-name amzn-s3-demo-bucket.s3.amazonaws.com \
    --default-root-object index.html
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EMLARXS9EXAMPLE",
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-11-22T00:55:15.705Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
```

```
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
```

```
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
```

```
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

Exemple 2 : pour créer une CloudFront distribution à l'aide d'un fichier JSON

L'`create-distribution`exemple suivant crée une distribution pour un compartiment S3 nomméamzn-s3-demo-bucket, et le spécifie également `index.html` comme objet racine par défaut, à l'aide d'un fichier JSON.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

Contenu de `dist-config.json` :

```
{
    "CallerReference": "cli-example",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
```

```
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
```

```
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

Voir l'exemple 1 pour un exemple de sortie.

Exemple 3 : pour créer une distribution CloudFront multi-locataires avec un certificat

L'`create-distribution`exemple suivant crée une CloudFront distribution prenant en charge plusieurs locataires et spécifie un certificat TLS.

```
aws cloudfront create-distribution \
```

```
    --distribution-config file://dist-config.json
```

Contenu de `dist-config.json` :

```
{
    "CallerReference": "cli-example-with-cert",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["HEAD", "GET"],
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["HEAD", "GET"]
            }
        }
    },
    "Enabled": true,
    "ViewerCertificate": {
        "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/191306a1-db01-49ca-90ef-fc414ee5dabc",
        "SSLSupportMethod": "sni-only"
    },
```

```
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E1HVIAU7UABC",
    "ETag": "E20LT7R1BABC",
    "Distribution": {
        "Id": "E1HVIAU7U12ABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E1HVIAU7U12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:33:31.117000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-with-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
```

```
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": ["HEAD", "GET"],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": ["HEAD", "GET"]
                    }
                },
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
```

```
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
                "GrpcConfig": {
                    "Enabled": false
                }
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": false,
                "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1.2_2021",
                "Certificate": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "CertificateSource": "acm"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
```

```
                        "Name": "tenantName",
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

Pour plus d'informations, consultez la section Travailler avec les distributions dans le manuel Amazon CloudFront Developer Guide.

Exemple 4 : pour créer une distribution CloudFront multi-locataires sans certificat

L'`create-distribution`exemple suivant crée une CloudFront distribution prenant en charge plusieurs locataires mais sans certificat TLS.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

Contenu de `dist-config.json` :

```
{
    "CallerReference": "cli-example",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
```

```
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        }
    },
    "Enabled": true,
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

Sortie :


```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E2GJ5J9QN12ABC",
    "ETag": "E37YLVVQIABC",
    "Distribution": {
        "Id": "E2GJ5J9QNABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E2GJ5J9QN12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:35:20.565000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-no-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
```

```
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
                "GrpcConfig": {
                    "Enabled": false
                }
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
```

```
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
                        "Name": "tenantName",
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

Pour plus d'informations, consultez Configurer les distributions dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDistributionà la section Référence des AWS CLI commandes.

`create-field-level-encryption-config`

L'exemple de code suivant montre comment utiliser`create-field-level-encryption-config`.

AWS CLI

Pour créer une configuration de CloudFront chiffrement au niveau du champ

L'exemple suivant crée une configuration de chiffrement au niveau du champ en fournissant les paramètres de configuration dans un fichier JSON nommé. `fle-config.json` Avant de créer une configuration de chiffrement au niveau du champ, vous devez disposer d'un profil de chiffrement au niveau du champ. Pour créer un profil, consultez la commande create-field-level-encryption -profile.

Pour plus d'informations sur le chiffrement CloudFront au niveau du champ, consultez la section Utilisation du chiffrement au niveau du champ pour protéger les données sensibles dans le manuel Amazon Developer Guide. CloudFront

```
aws cloudfront create-field-level-encryption-config \
    --field-level-encryption-config file://fle-config.json
```

Le fichier `fle-config.json` est un document JSON situé dans le dossier actuel qui contient les éléments suivants :

```
{
    "CallerReference": "cli-example",
    "Comment": "Example FLE configuration",
    "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
            "Quantity": 0
        }
    },
    "ContentTypeProfileConfig": {
        "ForwardWhenContentTypeIsUnknown": true,
```

```
        "ContentTypeProfiles": {
            "Quantity": 1,
            "Items": [
                {
                    "Format": "URLEncoded",
                    "ProfileId": "P280MFCLSYOCVU",
                    "ContentType": "application/x-www-form-urlencoded"
                }
            ]
        }
    }
}
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption/
C3KM2WVD605UAY",
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
```

```
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFieldLevelEncryptionConfigà la section Référence des AWS CLI commandes.

`create-field-level-encryption-profile`

L'exemple de code suivant montre comment utiliser`create-field-level-encryption-profile`.

AWS CLI

Pour créer un profil de CloudFront chiffrement au niveau du champ

L'exemple suivant crée un profil de chiffrement au niveau du champ en fournissant les paramètres dans un fichier JSON nommé. `fle-profile-config.json` Avant de créer un profil de chiffrement au niveau du champ, vous devez disposer d'une clé CloudFront publique. Pour créer une clé CloudFront publique, consultez la create-public-key commande.

Pour plus d'informations sur le chiffrement CloudFront au niveau du champ, consultez la section Utilisation du chiffrement au niveau du champ pour protéger les données sensibles dans le manuel Amazon Developer Guide. CloudFront

```
aws cloudfront create-field-level-encryption-profile \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

Le fichier `fle-profile-config.json` est un document JSON situé dans le dossier actuel qui contient les éléments suivants :

```
{
    "Name": "ExampleFLEProfile",
    "CallerReference": "cli-example",
    "Comment": "FLE profile for AWS CLI example",
    "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
            {
```

```
                "PublicKeyId": "K2K8NC4HVFE3M0",
                "ProviderId": "ExampleFLEProvider",
                "FieldPatterns": {
                    "Quantity": 1,
                    "Items": [
                        "ExampleSensitiveField"
                    ]
                }
            }
        ]
    }
}
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption-
profile/PPK0UOSIF5WSV",
    "ETag": "E2QWRUHEXAMPLE",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T01:03:16.537Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfile",
            "CallerReference": "cli-example",
            "Comment": "FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 1,
                            "Items": [
                                "ExampleSensitiveField"
                            ]
                        }
                    }
                ]
            }
        }
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFieldLevelEncryptionProfileà la section Référence des AWS CLI commandes.

`create-invalidation-for-distribution-tenant`

L'exemple de code suivant montre comment utiliser`create-invalidation-for-distribution-tenant`.

AWS CLI

Pour créer une CloudFront invalidation pour un locataire de distribution

L'`create-invalidation-for-distribution-tenant`exemple suivant crée une invalidation pour tous les fichiers d'un tenant CloudFront de distribution.

```
aws cloudfront create-invalidation-for-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJO1AB \
    --invalidation-batch '{"Paths": {"Quantity": 1, "Items": ["/*"]},
 "CallerReference": "invalidation-$(date +%s)"}'
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJO1AB/invalidation/I2JGL2F1ZAA426PGG0YLLKABC",
    "Invalidation": {
        "Id": "I2JGL2F1ZAA426PGG0YLLKABC",
        "Status": "InProgress",
        "CreateTime": "2025-05-07T16:59:25.947000+00:00",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 1,
                "Items": [
                    "/*"
                ]
            },
            "CallerReference": "invalidation-$(date +%s)"
        }
    }
```

```
}
```

Pour plus d'informations, consultez la section Invalider des fichiers pour supprimer du contenu dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateInvalidationForDistributionTenantà la section Référence des AWS CLI commandes.

`create-invalidation`

L'exemple de code suivant montre comment utiliser`create-invalidation`.

AWS CLI

Pour créer une invalidation pour une distribution CloudFront

L'`create-invalidation`exemple suivant crée une invalidation pour les fichiers spécifiés dans la CloudFront distribution spécifiée :

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",
    "Invalidation": {
        "Id": "I1JLWSDAP8FU89",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:24:51.407Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file2.png",
                    "/example-path/example-file.jpg"
                ]
            },
            "CallerReference": "cli-1575570291-670203"
        }
```

```
    }
}
```

Dans l'exemple précédent, la AWS CLI a automatiquement généré un résultat aléatoire`CallerReference`. Pour spécifier les vôtres `CallerReference` ou pour éviter de transmettre les paramètres d'invalidation en tant qu'arguments de ligne de commande, vous pouvez utiliser un fichier JSON. L'exemple suivant crée une invalidation pour deux fichiers, en fournissant les paramètres d'invalidation dans un fichier JSON nommé : `inv-batch.json`

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --invalidation-batch file://inv-batch.json
```

Contenu de `inv-batch.json` :

```
{
    "Paths": {
        "Quantity": 2,
        "Items": [
            "/example-path/example-file.jpg",
            "/example-path/example-file2.png"
        ]
    },
    "CallerReference": "cli-example"
}
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUYOIK",
    "Invalidation": {
        "Id": "I2J0I21PCUYOIK",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:40:49.413Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file2.png"
                ]
```

```
            },
            "CallerReference": "cli-example"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateInvalidationà la section Référence des AWS CLI commandes.

`create-public-key`

L'exemple de code suivant montre comment utiliser`create-public-key`.

AWS CLI

Pour créer une clé CloudFront publique

L'exemple suivant crée une clé CloudFront publique en fournissant les paramètres dans un fichier JSON nommépub-key-config.json. Avant de pouvoir utiliser cette commande, vous devez disposer d'une clé publique codée PEM. Pour plus d'informations, consultez la section Créer une paire de clés RSA dans le manuel Amazon CloudFront Developer Guide.

```
aws cloudfront create-public-key \
    --public-key-config file://pub-key-config.json
```

Le fichier pub-key-config.json est un document JSON dans le dossier actuel qui contient les éléments suivants. Notez que la clé publique est codée au format PEM.

```
{
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
    "Comment": "example public key"
}
```

Sortie :

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
        "Id": "KDFB19YGCR002",
        "CreatedTime": "2019-12-05T18:51:43.781Z",
        "PublicKeyConfig": {
            "CallerReference": "cli-example",
            "Name": "ExampleKey",
            "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePublicKeyà la section Référence des AWS CLI commandes.

`delete-cloud-front-origin-access-identity`

L'exemple de code suivant montre comment utiliser`delete-cloud-front-origin-access-identity`.

AWS CLI

Pour supprimer une identité CloudFront d'accès d'origine

L'exemple suivant supprime l'identité d'accès à l'origine (OAI) avec l'ID. `E74FTE3AEXAMPLE` Pour supprimer un OAI, vous devez avoir son identifiant et. `ETag` L'ID OAI est renvoyé dans la sortie des commandes -access-identity et create-cloud-front-origin -access-identity. list-cloud-front-origin Pour l'obtenir`ETag`, utilisez la commande get-cloud-front-origin -access-identity ou get-cloud-front-origin -. access-identity-config Utilisez l'`--if-match`option pour fournir les OAI. `ETag`

```
aws cloudfront delete-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

En cas de réussite, cette commande n'a aucune sortie.

- Pour plus de détails sur l'API, reportez-vous DeleteCloudFrontOriginAccessIdentityà la section Référence des AWS CLI commandes.

`delete-connection-group`

L'exemple de code suivant montre comment utiliser`delete-connection-group`.

AWS CLI

Pour supprimer un groupe de connexion

L'`delete-connection-group`exemple suivant supprime un groupe de connexions. Le groupe de connexion doit être désactivé et ne peut être associé à aucune CloudFront ressource.

```
aws cloudfront delete-connection-group \
    --id cg_2wjLpjbHkLUdhWAjHllcOeABC \
    --if-match ETVPDKIKX0DABC
```

En cas de réussite, cette commande n'a aucune sortie.

Pour plus d'informations sur la gestion des groupes de connexion, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteConnectionGroupà la section Référence des AWS CLI commandes.

`delete-distribution-tenant`

L'exemple de code suivant montre comment utiliser`delete-distribution-tenant`.

AWS CLI

Pour supprimer un tenant de distribution

L'`delete-distribution-tenant`exemple suivant supprime un tenant de distribution avec ETag `ETVPDKIKX0DABC`. Le tenant de distribution doit être désactivé et ne peut être associé à aucune CloudFront ressource.

```
aws cloudfront delete-distribution-tenant \
    --id dt_2wjMUbg3NHZEQ7OfoalP5zi1AB \
    --if-match ETVPDKIKX0DABC
```

En cas de réussite, cette commande n'a aucune sortie.

Pour plus d'informations, consultez Supprimer une distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDistributionTenantà la section Référence des AWS CLI commandes.

`delete-distribution`

L'exemple de code suivant montre comment utiliser`delete-distribution`.

AWS CLI

Pour supprimer une CloudFront distribution

L'exemple suivant supprime la CloudFront distribution avec l'ID. `EDFDVBD6EXAMPLE` Avant de pouvoir supprimer une distribution, vous devez la désactiver. Pour désactiver une distribution, utilisez la commande update-distribution. Pour plus d'informations, consultez les exemples de distribution de mises à jour.

Lorsqu'une distribution est désactivée, vous pouvez la supprimer. Pour supprimer une distribution, vous devez utiliser l'`--if-match`option permettant de fournir les informations de la distribution`ETag`. Pour obtenir le`ETag`, utilisez la commande get-distribution orget-distribution-config .

```
aws cloudfront delete-distribution \
    --id EDFDVBD6EXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

En cas de réussite, cette commande n'a aucune sortie.

- Pour plus de détails sur l'API, reportez-vous DeleteDistributionà la section Référence des AWS CLI commandes.

`delete-field-level-encryption-config`

L'exemple de code suivant montre comment utiliser`delete-field-level-encryption-config`.

AWS CLI

Pour supprimer une configuration de CloudFront chiffrement au niveau du champ

L'exemple suivant supprime la configuration de chiffrement CloudFront au niveau du champ avec l'ID. `C3KM2WVD605UAY` Pour supprimer une configuration de chiffrement au niveau du champ, vous devez disposer de son identifiant et. `ETag` L'ID est renvoyé dans la sortie des commandes create-field-level-encryption -config et list-field-level-encryption -configs. Pour l'obtenir`ETag`, utilisez la commande get-field-level-encryption or get-field-level-encryption -config. Utilisez l'`--if-match`option pour fournir les paramètres de configuration`ETag`.

```
aws cloudfront delete-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E26M4BIAV81ZF6
```

En cas de réussite, cette commande n'a aucune sortie.

- Pour plus de détails sur l'API, reportez-vous DeleteFieldLevelEncryptionConfigà la section Référence des AWS CLI commandes.

`delete-field-level-encryption-profile`

L'exemple de code suivant montre comment utiliser`delete-field-level-encryption-profile`.

AWS CLI

Pour supprimer un profil de CloudFront chiffrement au niveau du champ

L'exemple suivant supprime le profil de chiffrement CloudFront au niveau du champ avec l'ID. `PPK0UOSIF5WSV` Pour supprimer un profil de chiffrement au niveau du champ, vous devez disposer de son identifiant et. `ETag` L'ID est renvoyé dans la sortie des commandes create-field-level-encryption -profile et list-field-level-encryption -profiles. Pour l'obtenir`ETag`, utilisez la commande get-field-level-encryption -profile ou get-field-level-encryption -profile-config. Utilisez l'`--if-match`option pour fournir les informations du profil`ETag`.

```
aws cloudfront delete-field-level-encryption-profile \
```

```
    --id PPK0UOSIF5WSV \
    --if-match EJETYFJ9CL66D
```

En cas de réussite, cette commande n'a aucune sortie.

- Pour plus de détails sur l'API, reportez-vous DeleteFieldLevelEncryptionProfileà la section Référence des AWS CLI commandes.

`delete-public-key`

L'exemple de code suivant montre comment utiliser`delete-public-key`.

AWS CLI

Pour supprimer une clé CloudFront publique

L'exemple suivant supprime la clé CloudFront publique avec l'ID. `KDFB19YGCR002` Pour supprimer une clé publique, vous devez disposer de son identifiant et`ETag`. L'ID est renvoyé dans la sortie des list-public-keys commandes create-public-key et. Pour obtenir le`ETag`, utilisez la get-public-key-config commande get-public-key or. Utilisez l'`--if-match`option pour fournir les clés publiques`ETag`.

```
aws cloudfront delete-public-key \
    --id KDFB19YGCR002 \
    --if-match E2QWRUHEXAMPLE
```

En cas de réussite, cette commande n'a aucune sortie.

- Pour plus de détails sur l'API, reportez-vous DeletePublicKeyà la section Référence des AWS CLI commandes.

`disassociate-distribution-tenant-web-acl`

L'exemple de code suivant montre comment utiliser`disassociate-distribution-tenant-web-acl`.

AWS CLI

Pour dissocier une ACL Web d'un tenant de distribution

L'`disassociate-distribution-tenant-web-acl`exemple suivant dissocie une ACL Web d'un locataire de distribution avec ETag `E1PA6795UKMABC`.

```
aws cloudfront disassociate-distribution-tenant-web-acl \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB \
    --if-match E1PA6795UKMABC
```

Sortie :

```
{
    "ETag": "E13V1IB3VIYABC",
    "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB"
}
```

Pour plus d'informations, consultez la section Désactiver les protections de sécurité AWS WAF dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DisassociateDistributionTenantWebAclà la section Référence des AWS CLI commandes.

`disassociate-distribution-web-acl`

L'exemple de code suivant montre comment utiliser`disassociate-distribution-web-acl`.

AWS CLI

Pour dissocier une ACL Web d'une distribution CloudFront

L'`disassociate-distribution-web-acl`exemple suivant supprime l'association entre une ACL Web et une CloudFront distribution avec ETag `E13V1IB3VIYABC`.

```
aws cloudfront disassociate-distribution-web-acl \
    --id E1XNX8R2GOAABC \
    --if-match EEZQ9Z24VM1ABC
```

Sortie :

```
{
    "ETag": "E2YWS1C2J3OABC",
    "Id": "E1XNX8R2GOAABC"
}
```

Pour plus d'informations, consultez la section Désactiver les protections de sécurité AWS WAF dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DisassociateDistributionWebAclà la section Référence des AWS CLI commandes.

`get-cloud-front-origin-access-identity-config`

L'exemple de code suivant montre comment utiliser`get-cloud-front-origin-access-identity-config`.

AWS CLI

Pour obtenir une configuration d'identité CloudFront d'accès à l'origine

L'exemple suivant obtient des métadonnées relatives à l'identité CloudFront d'accès à l'origine (OAI) avec l'IDE74FTE3AEXAMPLE, y compris son`ETag`. L'ID OAI est renvoyé dans la sortie des commandes -access-identity et create-cloud-front-origin -access-identity. list-cloud-front-origin

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetCloudFrontOriginAccessIdentityConfigà la section Référence des AWS CLI commandes.

`get-cloud-front-origin-access-identity`

L'exemple de code suivant montre comment utiliser`get-cloud-front-origin-access-identity`.

AWS CLI

Pour obtenir une identité CloudFront d'accès à l'origine

L'exemple suivant obtient l'identité CloudFront d'accès à l'origine (OAI) avec l'IDE74FTE3AEXAMPLE, y compris son identifiant canonique S3 ETag et l'identifiant canonique S3 associé. L'ID OAI est renvoyé dans la sortie des commandes -access-identity et create-cloud-front-origin -access-identity. list-cloud-front-origin

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetCloudFrontOriginAccessIdentityà la section Référence des AWS CLI commandes.

`get-connection-group-by-routing-endpoint`

L'exemple de code suivant montre comment utiliserget-connection-group-by-routing-endpoint.

AWS CLI

Pour obtenir un groupe de connexion par point de terminaison de routage

L'get-connection-group-by-routing-endpointexemple suivant récupère des informations sur un groupe de connexion à l'aide de son point de terminaison de routage.

```
aws cloudfront get-connection-group-by-routing-endpoint \
    --routing-endpoint dvdg9gprgabc.cloudfront.net
```

Sortie :

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "Name": "connection-group-2",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
        "Ipv6Enabled": true,
        "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
        "Status": "Deployed",
        "Enabled": true,
        "IsDefault": false
    }
}
```

Pour plus d'informations, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetConnectionGroupByRoutingEndpointà la section Référence des AWS CLI commandes.

`get-connection-group`

L'exemple de code suivant montre comment utiliser`get-connection-group`.

AWS CLI

Pour obtenir un groupe CloudFront de connexion

L'`get-connection-group`exemple suivant permet de récupérer des informations sur un groupe de CloudFront connexion.

```
aws cloudfront get-connection-group \
    --identifier cg_2wjDWTBKTlRB87cAaUQFaakABC
```

Sortie :

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
```

```
        "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "Name": "connection-group-2",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
        "Ipv6Enabled": true,
        "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
        "Status": "Deployed",
        "Enabled": true,
        "IsDefault": false
    }
}
```

Pour plus d'informations, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetConnectionGroupà la section Référence des AWS CLI commandes.

`get-distribution-config`

L'exemple de code suivant montre comment utiliser`get-distribution-config`.

AWS CLI

Pour obtenir une configuration CloudFront de distribution

L'exemple suivant obtient les métadonnées relatives à la CloudFront distribution avec l'IDEDFDVBD6EXAMPLE, y compris son`ETag`. L'ID de distribution est renvoyé dans les commandes create-distribution et list-distributions.

```
aws cloudfront get-distribution-config \
    --id EDFDVBD6EXAMPLE
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
```

```
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
```

```
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
```

```
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDistributionConfigà la section Référence des AWS CLI commandes.

`get-distribution-tenant-by-domain`

L'exemple de code suivant montre comment utiliser`get-distribution-tenant-by-domain`.

AWS CLI

Pour obtenir des informations sur les locataires de distribution par domaine

L'`get-distribution-tenant-by-domain`exemple suivant récupère des informations sur un locataire de distribution utilisant le domaine spécifié.

```
aws cloudfront get-distribution-tenant-by-domain \
    --domain example.com
```

Sortie :

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "example-tenant-4",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
```

```
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "defaultValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-05-23T16:16:20.871000+00:00",
        "LastModifiedTime": "2025-05-23T16:16:20.871000+00:00",
        "Enabled": false,
        "Status": "Deployed"
    }
}
```

Pour plus d'informations, consultez Comprendre le fonctionnement des distributions multi-locataires dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDistributionTenantByDomainà la section Référence des AWS CLI commandes.

`get-distribution-tenant`

L'exemple de code suivant montre comment utiliser`get-distribution-tenant`.

AWS CLI

Pour obtenir des informations sur un locataire CloudFront de distribution

L'`get-distribution-tenant`exemple suivant permet de récupérer des informations sur un locataire CloudFront de distribution.

```
aws cloudfront get-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

Sortie :

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "example-tenant-2",
```

```
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "ConnectionGroupId": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:28.542000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:37.724000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

Pour plus d'informations, consultez Comprendre le fonctionnement des distributions multi-locataires dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDistributionTenantà la section Référence des AWS CLI commandes.

`get-distribution`

L'exemple de code suivant montre comment utiliser`get-distribution`.

AWS CLI

Pour obtenir une CloudFront distribution

L'`get-distribution`exemple suivant obtient la CloudFront distribution avec l'IDEDFDVBD6EXAMPLE, y compris son`ETag`. L'ID de distribution est renvoyé dans les commandes create-distribution et list-distributions.

```
aws cloudfront get-distribution \
    --id EDFDVBD6EXAMPLE
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
```

```
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "Deployed",
        "LastModifiedTime": "2019-12-04T23:35:41.433Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
```

```
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
```

```
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDistributionà la section Référence des AWS CLI commandes.

`get-field-level-encryption-config`

L'exemple de code suivant montre comment utiliser`get-field-level-encryption-config`.

AWS CLI

Pour obtenir des métadonnées relatives à une configuration de CloudFront chiffrement au niveau des champs

L'exemple suivant permet d'obtenir des métadonnées relatives à la configuration de CloudFront chiffrement au niveau des champs avec l'IDC3KM2WVD605UAY, y compris son : ETag

```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

Sortie :

```
{
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryptionConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example FLE configuration",
        "QueryArgProfileConfig": {
            "ForwardWhenQueryArgProfileIsUnknown": true,
            "QueryArgProfiles": {
                "Quantity": 0,
                "Items": []
            }
        },
        "ContentTypeProfileConfig": {
            "ForwardWhenContentTypeIsUnknown": true,
            "ContentTypeProfiles": {
                "Quantity": 1,
                "Items": [
                    {
                        "Format": "URLEncoded",
                        "ProfileId": "P280MFCLSYOCVU",
                        "ContentType": "application/x-www-form-urlencoded"
                    }
                ]
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFieldLevelEncryptionConfigà la section Référence des AWS CLI commandes.

`get-field-level-encryption-profile-config`

L'exemple de code suivant montre comment utiliser`get-field-level-encryption-profile-config`.

AWS CLI

Pour obtenir une configuration de CloudFront profil de chiffrement au niveau du champ

L'exemple suivant obtient les métadonnées relatives au profil de chiffrement CloudFront au niveau du champ avec IDPPK0UOSIF5WSV, y compris son : `ETag`

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0UOSIF5WSV
```

Sortie :

```
{
    "ETag": "E1QQG65FS2L2GC",
    "FieldLevelEncryptionProfileConfig": {
        "Name": "ExampleFLEProfile",
        "CallerReference": "cli-example",
        "Comment": "FLE profile for AWS CLI example",
        "EncryptionEntities": {
            "Quantity": 1,
            "Items": [
                {
                    "PublicKeyId": "K2K8NC4HVFE3M0",
                    "ProviderId": "ExampleFLEProvider",
                    "FieldPatterns": {
                        "Quantity": 1,
                        "Items": [
                            "ExampleSensitiveField"
                        ]
                    }
                }
            ]
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFieldLevelEncryptionProfileConfigà la section Référence des AWS CLI commandes.

`get-field-level-encryption-profile`

L'exemple de code suivant montre comment utiliser`get-field-level-encryption-profile`.

AWS CLI

Pour obtenir un profil de CloudFront chiffrement au niveau du champ

L'exemple suivant obtient le profil de chiffrement CloudFront au niveau du champ avec son IDPPK0UOSIF5WSV, y compris son : ETag

```
aws cloudfront get-field-level-encryption-profile --id PPK0UOSIF5WSV
```

Sortie :

```
{
    "ETag": "E1QQG65FS2L2GC",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T01:03:16.537Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfile",
            "CallerReference": "cli-example",
            "Comment": "FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 1,
                            "Items": [
                                "ExampleSensitiveField"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFieldLevelEncryptionProfileà la section Référence des AWS CLI commandes.

`get-field-level-encryption`

L'exemple de code suivant montre comment utiliser`get-field-level-encryption`.

AWS CLI

Pour obtenir une configuration de CloudFront chiffrement au niveau du champ

L'exemple suivant obtient la configuration de chiffrement CloudFront au niveau du champ avec l'IDC3KM2WVD605UAY, y compris son : ETag

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

Sortie :

```
{
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFieldLevelEncryptionà la section Référence des AWS CLI commandes.

`get-invalidation-for-distribution-tenant`

L'exemple de code suivant montre comment utiliser`get-invalidation-for-distribution-tenant`.

AWS CLI

Pour obtenir l'invalidation d'un locataire de distribution

L'`get-invalidation-for-distribution-tenant`exemple suivant permet d'obtenir des informations sur l'invalidation d'un locataire de distribution.

```
aws cloudfront get-invalidation-for-distribution-tenant \
    --distribution-tenant-id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB \
    --id I4CU23QAPKMUDUU06F9OFGFABC
```

Sortie :

```
{
    "Invalidation": {
        "Id": "I4CU23QAPKMUDUU06F9OFGFABC",
        "Status": "Completed",
        "CreateTime": "2025-05-06T15:46:12.824000+00:00",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example/invalidation",
                    "/more/invalidations"
                ]
            },
            "CallerReference": "007ee5a6-d0a0-42be-bb61-e7b915969b48"
        }
    }
}
```

Pour plus d'informations, consultez la section Invalider des fichiers pour supprimer du contenu dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetInvalidationForDistributionTenantà la section Référence des AWS CLI commandes.

`get-invalidation`

L'exemple de code suivant montre comment utiliser`get-invalidation`.

AWS CLI

Pour obtenir une CloudFront invalidation

L'exemple suivant obtient l'invalidation avec l'ID `I2J0I21PCUYOIK` de la CloudFront distribution avec l'ID : `EDFDVBD6EXAMPLE`

```
aws cloudfront get-invalidation --id I2J0I21PCUYOIK --distribution-
id EDFDVBD6EXAMPLE
```

Sortie :

```
{
    "Invalidation": {
        "Status": "Completed",
        "InvalidationBatch": {
            "Paths": {
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file-2.jpg"
                ],
                "Quantity": 2
            },
            "CallerReference": "cli-example"
        },
        "Id": "I2J0I21PCUYOIK",
        "CreateTime": "2019-12-05T18:40:49.413Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetInvalidationà la section Référence des AWS CLI commandes.

`get-managed-certificate-details`

L'exemple de code suivant montre comment utiliser`get-managed-certificate-details`.

AWS CLI

Pour obtenir les détails des certificats gérés

L'`get-managed-certificate-details`exemple suivant récupère les détails d'un certificat ACM CloudFront géré.

```
aws cloudfront get-managed-certificate-details \
    --identifier dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

Sortie :

```
{
    "ManagedCertificateDetails": {
        "CertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/655dc1fe-6d37-451d-a013-c2db3a034abc",
        "CertificateStatus": "pending-validation",
        "ValidationTokenHost": "self-hosted",
        "ValidationTokenDetails": [
            {
                "Domain": "example.com",
                "RedirectTo": "validation.us-east-1.acm-
validations.aws/123456789012/.well-known/pki-validation/
b315c9ae21284e7918bb9f3f422ab1c7.txt",
                "RedirectFrom": "example.com/.well-known/pki-validation/
b315c9ae21284e7918bb9f3f422ac3c7.txt"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Demander des certificats pour votre client CloudFront de distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetManagedCertificateDetailsà la section Référence des AWS CLI commandes.

`get-public-key-config`

L'exemple de code suivant montre comment utiliser`get-public-key-config`.

AWS CLI

Pour obtenir une configuration par clé CloudFront publique

L'exemple suivant obtient les métadonnées relatives à la clé CloudFront publique associée à l'IDKDFB19YGCR002, y compris sonETag. L'ID de clé publique est renvoyé dans les list-public-keys commandes create-public-key et.

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKeyConfig": {
        "CallerReference": "cli-example",
        "Name": "ExampleKey",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "Comment": "example public key"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPublicKeyConfigà la section Référence des AWS CLI commandes.

`get-public-key`

L'exemple de code suivant montre comment utiliserget-public-key.

AWS CLI

Pour obtenir une clé CloudFront publique

L'exemple suivant obtient la clé CloudFront publique avec l'IDKDFB19YGCR002, y compris sonETag. L'ID de clé publique est renvoyé dans les list-public-keys commandes create-public-key et.

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
        "Id": "KDFB19YGCR002",
        "CreatedTime": "2019-12-05T18:51:43.781Z",
        "PublicKeyConfig": {
            "CallerReference": "cli-example",
            "Name": "ExampleKey",
            "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPublicKeyà la section Référence des AWS CLI commandes.

`list-cloud-front-origin-access-identities`

L'exemple de code suivant montre comment utiliser`list-cloud-front-origin-access-identities`.

AWS CLI

Pour répertorier les identités CloudFront d'accès à l'origine

L'exemple suivant permet d'obtenir une liste des identités CloudFront d'accès d'origine (OAIs) de votre AWS compte :

```
aws cloudfront list-cloud-front-origin-access-identities
```

Sortie :

```
{
    "CloudFrontOriginAccessIdentityList": {
        "Items": [
            {
                "Id": "E74FTE3AEXAMPLE",
                "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
                "Comment": "Example OAI"
            },
            {
                "Id": "EH1HDMBEXAMPLE",
                "S3CanonicalUserId":
 "1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
                "Comment": "Test OAI"
            },
            {
                "Id": "E2X2C9TEXAMPLE",
                "S3CanonicalUserId":
 "cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
                "Comment": "Example OAI #2"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListCloudFrontOriginAccessIdentitiesà la section Référence des AWS CLI commandes.

`list-connection-groups`

L'exemple de code suivant montre comment utiliser`list-connection-groups`.

AWS CLI

Pour répertorier les groupes de connexions

L'`list-connection-groups`exemple suivant répertorie les groupes de connexion disponibles dans votre AWS compte.

```
aws cloudfront list-connection-groups
```

Sortie :

```
{
    "ConnectionGroups": [
        {
            "Id": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Name": "CreatedByCloudFront-cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "RoutingEndpoint": "d3sx0pso7m5abc.cloudfront.net",
            "CreatedTime": "2025-05-05T22:32:29.630000+00:00",
            "LastModifiedTime": "2025-05-05T22:32:29.630000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": true,
            "Status": "Deployed",
            "IsDefault": true
        },
        {
            "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "Name": "connection-group-2",
            "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
            "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
            "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": true,
            "Status": "Deployed",
            "IsDefault": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListConnectionGroupsà la section Référence des AWS CLI commandes.

`list-distribution-tenants-by-customization`

L'exemple de code suivant montre comment utiliser`list-distribution-tenants-by-customization`.

AWS CLI

Pour répertorier les locataires de distribution par personnalisation

L'`list-distribution-tenants-by-customization`exemple suivant répertorie les locataires de distribution qui utilisent l'ACL Web spécifiée.

```
aws cloudfront list-distribution-tenants-by-customization \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/
CreatedByCloudFront-0273cd2f/a3c19bce-42b5-48a1-a8d4-b2bb2f28eabc
```

Sortie :

```
{
    "DistributionTenantList": [
        {
            "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "example-tenant-2",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
            "Domains": [
                {
                    "Domain": "example.com",
                    "Status": "inactive"
                }
            ],
            "ConnectionGroupId": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "Customizations": {
                "WebAcl": {
                    "Action": "override",
                    "Arn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/
CreatedByCloudFront-0273cd2f/a3c19bce-42b5-48a1-a8d4-b2bb2f28eabc"
                },
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "AL"
```

```
                    ]
                }
            },
            "CreatedTime": "2025-05-06T15:42:28.542000+00:00",
            "LastModifiedTime": "2025-05-06T16:14:08.710000+00:00",
            "ETag": "E1F83G8C2ARABC",
            "Enabled": true,
            "Status": "Deployed"
        }
    ]
}
```

Pour plus d'informations, consultez la section Personnalisations des locataires de distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDistributionTenantsByCustomizationà la section Référence des AWS CLI commandes.

`list-distribution-tenants`

L'exemple de code suivant montre comment utiliser`list-distribution-tenants`.

AWS CLI

Pour répertorier les locataires CloudFront de distribution

L'`list-distribution-tenants`exemple suivant répertorie 3 locataires CloudFront de distribution de votre AWS compte par groupe de connexion associé.

```
aws cloudfront list-distribution-tenants \
    --association-filter ConnectionGroupId=cg_2whCJoXMYCjHcxaLGrkllvyABC \
    --max-items 3
```

Sortie :

```
{
    "DistributionTenantList": [
        {
            "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant-customizations",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMvQgam3QkJo2z54FDl91dk1AB",
```

```
            "Domains": [
                {
                    "Domain": "example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Customizations": {
                "WebAcl": {
                    "Action": "disable"
                },
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "DE"
                    ]
                }
            },
            "CreatedTime": "2025-06-11T15:54:02.142000+00:00",
            "LastModifiedTime": "2025-06-11T15:54:02.142000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        },
        {
            "Id": "dt_2yMuV7NJuBcAB0cwwxMCBZQ1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMuV7NJuBcAB0cwwxMCBZQ1AB",
            "Domains": [
                {
                    "Domain": "1.example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Customizations": {
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "DE"
                    ]
                }
```

```
            },
            "CreatedTime": "2025-06-11T15:46:23.466000+00:00",
            "LastModifiedTime": "2025-06-11T15:46:23.466000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        },
        {
            "Id": "dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant-2",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
            "Domains": [
                {
                    "Domain": "2.example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "CreatedTime": "2025-05-23T16:16:20.871000+00:00",
            "LastModifiedTime": "2025-05-23T16:16:20.871000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        }
    ],
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}
```

Pour plus d'informations, consultez Comprendre le fonctionnement des distributions multi-locataires dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDistributionTenantsà la section Référence des AWS CLI commandes.

`list-distributions-by-connection-mode`

L'exemple de code suivant montre comment utiliser`list-distributions-by-connection-mode`.

AWS CLI

Pour répertorier CloudFront les distributions par mode de connexion

L'`list-distributions-by-connection-mode`exemple suivant répertorie les CloudFront distributions avec le mode de connexion spécifié.

```
aws cloudfront list-distributions-by-connection-mode \
    --connection-mode tenant-only
```

Sortie :

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E1XNX8R2GOAABC",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1XNX8R2GOAABC",
                "ETag": "EPT4JPJQDY1ABC",
                "Status": "Deployed",
                "LastModifiedTime": "2025-05-23T16:16:15.691000+00:00",
                "DomainName": "-",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "example-cfn-simple-distribution123",
                            "DomainName": "example.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
```

```
                    },
                    "OriginAccessControlId": "E2CJRMB5LKEABC"
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "example-cfn-simple-distribution123",
            "TrustedKeyGroups": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "Compress": true,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": "",
            "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5abc",
            "GrpcConfig": {
                "Enabled": false
            }
        },
        "CacheBehaviors": {
            "Quantity": 0
```

```
                },
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": false,
                    "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/ec53f564-ea5a-4e4a-a0a2-e3c989449abc",
                    "SSLSupportMethod": "sni-only",
                    "MinimumProtocolVersion": "TLSv1.2_2021",
                    "Certificate": "arn:aws:acm:us-east-1:123456789012:certificate/
ec53f564-ea5a-4e4a-a0a2-e3c989449abc",
                    "CertificateSource": "acm"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
                        "Quantity": 0
                    }
                },
                "WebACLId": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-
global-example/626900da-5f64-418b-ba9b-743f3746cabc",
                "HttpVersion": "http2",
                "IsIPV6Enabled": false,
                "Staging": false,
                "ConnectionMode": "tenant-only"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDistributionsByConnectionModeà la section Référence des AWS CLI commandes.

`list-distributions`

L'exemple de code suivant montre comment utiliser`list-distributions`.

AWS CLI

Pour répertorier CloudFront les distributions

L'exemple suivant permet d'obtenir la liste des CloudFront distributions de votre AWS compte.

```
aws cloudfront list-distributions
```

Sortie :

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E23YS8OEXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E23YS8OEXAMPLE",
                "Status": "Deployed",
                "LastModifiedTime": "2024-08-05T18:23:40.375000+00:00",
                "DomainName": "abcdefgh12ijk.cloudfront.net",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                            "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
                            },
                            "OriginAccessControlId": "EIAP8PEXAMPLE"
```

```
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                    "TrustedSigners": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "TrustedKeyGroups": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ],
                        "CachedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "SmoothStreaming": false,
                    "Compress": true,
                    "LambdaFunctionAssociations": {
                        "Quantity": 0
                    },
                    "FunctionAssociations": {
                        "Quantity": 0
                    },
                    "FieldLevelEncryptionId": "",
                    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e886EXAMPLE"
                },
                "CacheBehaviors": {
```

```
                    "Quantity": 0
                },
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": true,
                    "SSLSupportMethod": "vip",
                    "MinimumProtocolVersion": "TLSv1",
                    "CertificateSource": "cloudfront"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
                        "Quantity": 0
                    }
                },
                "WebACLId": "",
                "HttpVersion": "HTTP2",
                "IsIPV6Enabled": true,
                "Staging": false
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListDistributionsà la section Référence des AWS CLI commandes.

`list-domain-conflicts`

L'exemple de code suivant montre comment utiliser`list-domain-conflicts`.

AWS CLI

Pour répertorier les conflits de domaine relatifs à une CloudFront distribution

L'`list-domain-conflicts`exemple suivant répertorie les conflits de domaine relatifs à une CloudFront distribution.

```
aws cloudfront list-domain-conflicts \
    --domain example.com \
    --domain-control-validation-
resource "DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8J1AB"
```

Sortie :

```
{
    "DomainConflicts": [
        {
            "Domain": "example.com",
            "ResourceType": "distribution-tenant",
            "ResourceId": "***************ohWzv1b9It8J1AB",
            "AccountId": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez la section Déplacer un autre nom de domaine vers une autre distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDomainConflictsà la section Référence des AWS CLI commandes.

`list-field-level-encryption-configs`

L'exemple de code suivant montre comment utiliser`list-field-level-encryption-configs`.

AWS CLI

Pour répertorier les configurations de chiffrement CloudFront au niveau des champs

L'exemple suivant permet d'obtenir une liste des configurations de chiffrement CloudFront au niveau des champs de votre AWS compte :

```
aws cloudfront list-field-level-encryption-configs
```

Sortie :

```
{
```

```
    "FieldLevelEncryptionList": {
        "MaxItems": 100,
        "Quantity": 1,
        "Items": [
            {
                "Id": "C3KM2WVD605UAY",
                "LastModifiedTime": "2019-12-10T21:30:18.974Z",
                "Comment": "Example FLE configuration",
                "QueryArgProfileConfig": {
                    "ForwardWhenQueryArgProfileIsUnknown": true,
                    "QueryArgProfiles": {
                        "Quantity": 0,
                        "Items": []
                    }
                },
                "ContentTypeProfileConfig": {
                    "ForwardWhenContentTypeIsUnknown": true,
                    "ContentTypeProfiles": {
                        "Quantity": 1,
                        "Items": [
                            {
                                "Format": "URLEncoded",
                                "ProfileId": "P280MFCLSYOCVU",
                                "ContentType": "application/x-www-form-urlencoded"
                            }
                        ]
                    }
                }
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListFieldLevelEncryptionConfigsà la section Référence des AWS CLI commandes.

`list-field-level-encryption-profiles`

L'exemple de code suivant montre comment utiliser`list-field-level-encryption-profiles`.

AWS CLI

Pour répertorier les profils de chiffrement CloudFront au niveau des champs

L'exemple suivant permet d'obtenir la liste des profils de chiffrement CloudFront au niveau des champs de votre AWS compte :

```
aws cloudfront list-field-level-encryption-profiles
```

Sortie :

```
{
    "FieldLevelEncryptionProfileList": {
        "MaxItems": 100,
        "Quantity": 2,
        "Items": [
            {
                "Id": "P280MFCLSYOCVU",
                "LastModifiedTime": "2019-12-05T01:05:39.896Z",
                "Name": "ExampleFLEProfile",
                "EncryptionEntities": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "PublicKeyId": "K2K8NC4HVFE3M0",
                            "ProviderId": "ExampleFLEProvider",
                            "FieldPatterns": {
                                "Quantity": 1,
                                "Items": [
                                    "ExampleSensitiveField"
                                ]
                            }
                        }
                    ]
                },
                "Comment": "FLE profile for AWS CLI example"
            },
            {
                "Id": "PPK0UOSIF5WSV",
                "LastModifiedTime": "2019-12-10T01:03:16.537Z",
                "Name": "ExampleFLEProfile2",
                "EncryptionEntities": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "PublicKeyId": "K2ABC10EXAMPLE",
                            "ProviderId": "ExampleFLEProvider2",
```

```
                            "FieldPatterns": {
                                "Quantity": 1,
                                "Items": [
                                    "ExampleSensitiveField2"
                                ]
                            }
                        }
                    ]
                },
                "Comment": "FLE profile #2 for AWS CLI example"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListFieldLevelEncryptionProfilesà la section Référence des AWS CLI commandes.

`list-invalidations-for-distribution-tenant`

L'exemple de code suivant montre comment utiliser`list-invalidations-for-distribution-tenant`.

AWS CLI

Pour répertorier les invalidations d'un locataire de distribution CloudFront

L'`list-invalidations-for-distribution-tenant`exemple suivant répertorie les invalidations d'un locataire de CloudFront distribution.

```
aws cloudfront list-invalidations-for-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

Sortie :

```
{
    "InvalidationList": {
        "Items": [
            {
                "Id": "I4CU23QAPKMUDUU06F9OFGFABC",
                "CreateTime": "2025-05-06T15:46:12.824000+00:00",
                "Status": "Completed"
```

```
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Invalider des fichiers pour supprimer du contenu dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListInvalidationsForDistributionTenantà la section Référence des AWS CLI commandes.

`list-invalidations`

L'exemple de code suivant montre comment utiliser`list-invalidations`.

AWS CLI

Pour répertorier les CloudFront invalidations

L'exemple suivant permet d'obtenir une liste des invalidations pour la CloudFront distribution avec l'ID : EDFDVBD6EXAMPLE

```
aws cloudfront list-invalidations --distribution-id EDFDVBD6EXAMPLE
```

Sortie :

```
{
    "InvalidationList": {
        "Marker": "",
        "Items": [
            {
                "Status": "Completed",
                "Id": "YNY2LI2BVJ4NJU",
                "CreateTime": "2019-08-31T21:15:52.042Z"
            }
        ],
        "IsTruncated": false,
        "MaxItems": 100,
        "Quantity": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListInvalidationsà la section Référence des AWS CLI commandes.

`list-public-keys`

L'exemple de code suivant montre comment utiliser`list-public-keys`.

AWS CLI

Pour répertorier CloudFront les clés publiques

L'exemple suivant permet d'obtenir la liste des clés CloudFront publiques de votre AWS compte :

```
aws cloudfront list-public-keys
```

Sortie :

```
{
    "PublicKeyList": {
        "MaxItems": 100,
        "Quantity": 2,
        "Items": [
            {
                "Id": "K2K8NC4HVFE3M0",
                "Name": "ExampleKey",
                "CreatedTime": "2019-12-05T01:04:28.818Z",
                "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
                "Comment": "example public key"
            },
            {
                "Id": "K1S0LWQ2L5HTBU",
                "Name": "ExampleKey2",
                "CreatedTime": "2019-12-09T23:28:11.110Z",
                "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApOCAg88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1gol7FV6pYNl78iJg3EO8JBbwtlH
```

```
+cR9\nLGSf60NDeVhm76Oc39Np/vWgOdsGQcRbi9WmKZeSODqjQGzVZWqPmito3FzWVk6b
\nfVY5N36U/RdbVAJm95Km+qaMYlbIdF40t72bi3IkKYV5hlB2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmmgIVnkOQM37NYeoDnkaOuTpu\nha/
+3b8tOb2z3LBVHPkp85zJRAOXacSwf5rZtPYKBNFsixTa2n55k2r218mOkMC4\nUwIDAQAB\n-----END
 PUBLIC KEY-----",
                "Comment": "example public key #2"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListPublicKeysà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags d'une CloudFront distribution

L'exemple suivant permet d'obtenir la liste des balises d'une CloudFront distribution :

```
aws cloudfront list-tags-for-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE
```

Sortie :

```
{
    "Tags": {
        "Items": [
            {
                "Key": "DateCreated",
                "Value": "2019-12-04"
            },
            {
                "Key": "Name",
                "Value": "Example name"
            },
            {
                "Key": "Project",
```

```
                "Value": "Example project"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`sign`

L'exemple de code suivant montre comment utiliser`sign`.

AWS CLI

Pour signer une CloudFront URL

L'exemple suivant signe une CloudFront URL. Pour signer une URL, vous avez besoin de l'ID de paire de clés (appelé ID de clé d'accès dans la console de AWS gestion) et de la clé privée de la paire de CloudFront clés du signataire de confiance. Pour plus d'informations sur les cookies signés URLs, consultez la section Diffuser du contenu privé avec des cookies signés URLs et signés dans le manuel Amazon CloudFront Developer Guide.

```
aws cloudfront sign \
    --url https://d111111abcdef8.cloudfront.net/private-content/private-file.html \
    --key-pair-id APKAEIBAERJR2EXAMPLE \
    --private-key file://cf-signer-priv-key.pem \
    --date-less-than 2020-01-01
```

Sortie :

```
https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKVc6pwkif6oZc-
JWSpDkH0UH7EBGGqvgurkecCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoeenH0GFteRf9ELK-
Bs3nljTLjtbgzIUt7QJNKXcWr8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GGl6l2kgCpXUB6Jx6K~Y3kpVOdzUPOIqFLHAnJojbhxqrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FKOqNEaWDIB
GNvjRJxqWf93uMobeMOiVYahb-e0KItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE
```

- Pour plus de détails sur l'API, consultez la section Référence des AWS CLI commandes de connexion.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une CloudFront distribution

L'`tag-resource`exemple suivant ajoute deux balises à la CloudFront distribution spécifiée.

```
aws cloudfront tag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example
 project"}]'
```

Au lieu d'utiliser des arguments de ligne de commande, vous pouvez fournir les balises dans un fichier JSON, comme illustré dans l'exemple suivant :

```
aws cloudfront tag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tags file://tags.json
```

Contenu de `tags.json` :

```
{
    "Items": [
        {
            "Key": "Name",
            "Value": "Example name"
        },
        {
            "Key": "Project",
            "Value": "Example project"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une CloudFront distribution

L'exemple suivant supprime deux balises d'une CloudFront distribution à l'aide d'arguments de ligne de commande :

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys Items=Name,Project
```

Au lieu d'utiliser des arguments de ligne de commande, vous pouvez fournir les clés de balise dans un fichier JSON, comme illustré dans l'exemple suivant :

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys file://tag-keys.json
```

Le fichier `tag-keys.json` est un document JSON situé dans le dossier actuel qui contient les éléments suivants :

```
{
    "Items": [
        "Name",
        "Project"
    ]
}
```

En cas de réussite, cette commande n'a aucune sortie.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-cloud-front-origin-access-identity`

L'exemple de code suivant montre comment utiliser`update-cloud-front-origin-access-identity`.

AWS CLI

Pour mettre à jour une identité CloudFront d'accès d'origine

L'exemple suivant met à jour l'identité d'accès à l'origine (OAI) avec l'IDE74FTE3AEXAMPLE. Le seul champ que vous pouvez mettre à jour est celui de l'OAI. Comment

Pour mettre à jour un OAI, vous devez avoir l'identifiant de l'OAI et. ETag L'ID OAI est renvoyé dans la sortie des commandes -access-identity et create-cloud-front-origin -access-identity. list-cloud-front-origin Pour l'obtenirETag, utilisez la commande get-cloud-front-origin -access-identity ou get-cloud-front-origin -. access-identity-config Utilisez l'--if-matchoption pour fournir les OAI. ETag

```
aws cloudfront update-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --cloud-front-origin-access-identity-config \
        CallerReference=cli-example,Comment="Example OAI Updated"
```

Vous pouvez accomplir la même chose en fournissant la configuration OAI dans un fichier JSON, comme illustré dans l'exemple suivant :

```
aws cloudfront update-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --cloud-front-origin-access-identity-config file://OAI-config.json
```

Le fichier OAI-config.json est un document JSON situé dans le répertoire actuel qui contient les éléments suivants :

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI Updated"
}
```

Que vous fournissiez la configuration OAI avec un argument de ligne de commande ou un fichier JSON, le résultat est le même :

```
{
    "ETag": "E9LHASXEXAMPLE",
```

```
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI Updated"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateCloudFrontOriginAccessIdentityà la section Référence des AWS CLI commandes.

`update-connection-group`

L'exemple de code suivant montre comment utiliser`update-connection-group`.

AWS CLI

Pour mettre à jour un groupe CloudFront de connexion

L'`update-connection-group`exemple suivant désactive et désactive un groupe de CloudFront connexions. IPv6

```
aws cloudfront update-connection-group \
    --id cg_2yHsDkcPKeUlVkk3aEgLKcjABC \
    --no-ipv6-enabled \
    --no-enabled \
    --if-match E3UN6WX5RRO2ABC
```

Sortie :

```
{
    "ETag": "E1F83G8C2ARABC",
    "ConnectionGroup": {
        "Id": "cg_2yHsDkcPKeUlVkk3aEgLKcjABC",
        "Name": "cg-example",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2yHsDkcPKeUlVkk3aEgLKcjABC",
        "CreatedTime": "2025-06-09T20:58:35.481000+00:00",
        "LastModifiedTime": "2025-06-11T16:25:54.280000+00:00",
```

```
        "Ipv6Enabled": false,
        "RoutingEndpoint": "du9xp1elo1abc.cloudfront.net",
        "Status": "InProgress",
        "Enabled": false,
        "IsDefault": false
    }
}
```

Pour plus d'informations, consultez la section Créer un groupe de connexion personnalisé (facultatif) dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateConnectionGroupà la section Référence des AWS CLI commandes.

`update-distribution-tenant`

L'exemple de code suivant montre comment utiliser`update-distribution-tenant`.

AWS CLI

Pour mettre à jour un tenant CloudFront de distribution

L'`update-distribution-tenant`exemple suivant met à jour un tenant de CloudFront distribution avec une nouvelle valeur de paramètre et ajoute un pays aux restrictions géographiques.

```
aws cloudfront update-distribution-tenant \
    --cli-input-json file://update-tenant.json
```

Contenu de `update-tenant.json` :

```
{
    "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
    "IfMatch": "E1F83G8C2ARABC",
    "Parameters": [
        {
            "Name": "testParam",
            "Value": "newParameterValue"
        }
    ],
    "Customizations": {
        "WebAcl": {
```

```
            "Action": "disable"
        },
        "GeoRestrictions": {
            "RestrictionType": "whitelist",
            "Locations": [
                "DE",
                "GB",
                "ES"
            ]
        }
    }
}
```

Sortie :

```
{
    "ETag": "E1PA6795UKMABC",
    "DistributionTenant": {
        "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "new-tenant-customizations",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMvQgam3QkJo2z54FDl91dk1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Customizations": {
            "WebAcl": {
                "Action": "disable"
            },
            "GeoRestrictions": {
                "RestrictionType": "whitelist",
                "Locations": [
                    "DE",
                    "ES",
                    "GB"
                ]
            }
        },
        "Parameters": [
```

```
            {
                "Name": "testParam",
                "Value": "newParameterValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-06-11T15:54:02.142000+00:00",
        "LastModifiedTime": "2025-06-11T16:42:45.531000+00:00",
        "Enabled": false,
        "Status": "InProgress"
    }
}
```

Pour plus d'informations, consultez la section Personnalisations des locataires de distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDistributionTenantà la section Référence des AWS CLI commandes.

`update-distribution`

L'exemple de code suivant montre comment utiliser`update-distribution`.

AWS CLI

Exemple 1 : pour mettre à jour l'objet racine par défaut d'une CloudFront distribution

L'exemple suivant met à jour l'objet racine par défaut `index.html` pour la CloudFront distribution avec l'IDEDFDVBD6EXAMPLE.

```
aws cloudfront update-distribution \
    --id EDFDVBD6EXAMPLE \
    --default-root-object index.html
```

Sortie :

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:55:39.870Z",
```

```
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "example-website",
                        "DomainName": "www.example.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "CustomOriginConfig": {
                            "HTTPPort": 80,
                            "HTTPSPort": 443,
                            "OriginProtocolPolicy": "match-viewer",
                            "OriginSslProtocols": {
                                "Quantity": 2,
                                "Items": [
                                    "SSLv3",
                                    "TLSv1"
                                ]
                            },
                            "OriginReadTimeout": 30,
                            "OriginKeepaliveTimeout": 5
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "example-website",
```

```
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 1,
                        "Items": [
                            "*"
                        ]
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
```

```
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http1.1",
            "IsIPV6Enabled": true
        }
    }
}
```

Exemple 2 : pour mettre à jour une CloudFront distribution

L'exemple suivant désactive la CloudFront distribution avec l'ID `EMLARXS9EXAMPLE` en fournissant la configuration de distribution dans un fichier JSON nommé`dist-config-disable.json`. Pour mettre à jour une distribution, vous devez utiliser l'`--if-match`option permettant de fournir les informations de la distribution`ETag`. Pour obtenir le`ETag`, utilisez la commande get-distribution or get-distribution-config. Notez que le `Enabled` champ est défini sur `false` dans le fichier JSON.

Après avoir utilisé l'exemple suivant pour désactiver une distribution, vous pouvez utiliser la commande delete-distribution pour la supprimer.

```
aws cloudfront update-distribution \
    --id EMLARXS9EXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --distribution-config file://dist-config-disable.json
```

Contenu de `dist-config-disable.json` :

```
{
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
```

```
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
```

```
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": false,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

Sortie :

```
{
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:32:35.553Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1574382155-496510",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
```

```
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
```

```
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": false,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
```

```
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateDistributionà la section Référence des AWS CLI commandes.

`update-domain-association`

L'exemple de code suivant montre comment utiliser`update-domain-association`.

AWS CLI

Pour mettre à jour une association de domaines

L'`update-domain-association`exemple suivant met à jour une association de domaine pour un locataire de distribution avec ETag `E23ZP02F085ABC`.

```
aws cloudfront update-domain-association \
    --domain example.com \
    --target-resource DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8J1AB \
    --if-match E23ZP02F085ABC
```

Sortie :

```
{
    "ETag": "ETVPDKIKX0ABC",
    "Domain": "example.com",
    "ResourceId": "dt_2x9GhoK0TZRsohWzv1b9It8J1AB"
}
```

Pour plus d'informations, consultez la section Déplacer un autre nom de domaine vers une autre distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainAssociationà la section Référence des AWS CLI commandes.

`update-field-level-encryption-config`

L'exemple de code suivant montre comment utiliser`update-field-level-encryption-config`.

AWS CLI

Pour mettre à jour une configuration de chiffrement CloudFront au niveau des champs

L'exemple suivant met à jour le `Comment` champ de la configuration de chiffrement au niveau du champ avec l'ID `C3KM2WVD605UAY` en fournissant les paramètres dans un fichier JSON.

Pour mettre à jour une configuration de chiffrement au niveau du champ, vous devez disposer de l'ID de la configuration et. `ETag` L'ID est renvoyé dans la sortie des commandes create-field-level-encryption -config et list-field-level-encryption -configs. Pour l'obtenir`ETag`, utilisez la commande get-field-level-encryption or get-field-level-encryption -config. Utilisez l'`--if-match`option pour fournir les paramètres de configuration`ETag`.

```
aws cloudfront update-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E2P4Z4VU7TY5SG \
    --field-level-encryption-config file://fle-config.json
```

Le fichier `fle-config.json` est un document JSON situé dans le répertoire actuel qui contient les éléments suivants :

```
{
    "CallerReference": "cli-example",
    "Comment": "Updated example FLE configuration",
    "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
            "Quantity": 0
        }
    },
    "ContentTypeProfileConfig": {
        "ForwardWhenContentTypeIsUnknown": true,
        "ContentTypeProfiles": {
            "Quantity": 1,
            "Items": [
                {
                    "Format": "URLEncoded",
                    "ProfileId": "P280MFCLSYOCVU",
```

```
                    "ContentType": "application/x-www-form-urlencoded"
                }
            ]
        }
    }
}
```

Sortie :

```
{
    "ETag": "E26M4BIAV81ZF6",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T22:26:26.170Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Updated example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateFieldLevelEncryptionConfigà la section Référence des AWS CLI commandes.

update-field-level-encryption-profile

L'exemple de code suivant montre comment utiliserupdate-field-level-encryption-profile.

AWS CLI

Pour mettre à jour un profil de chiffrement CloudFront au niveau du champ

L'exemple suivant met à jour le profil de chiffrement au niveau du champ avec l'ID. PPK0UOSIF5WSV Cet exemple met à jour le Name et du profilComment, et ajoute un deuxième FieldPatterns élément, en fournissant les paramètres dans un fichier JSON.

Pour mettre à jour un profil de chiffrement au niveau du champ, vous devez disposer de l'ID du profil et. ETag L'ID est renvoyé dans la sortie des commandes create-field-level-encryption -profile et list-field-level-encryption -profiles. Pour l'obtenirETag, utilisez la commande get-field-level-encryption -profile ou get-field-level-encryption -profile-config. Utilisez l'--if-matchoption pour fournir les informations du profilETag.

```
aws cloudfront update-field-level-encryption-profile \
    --id PPK0UOSIF5WSV \
    --if-match E1QQG65FS2L2GC \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

Le fichier fle-profile-config.json est un document JSON situé dans le répertoire actuel qui contient les éléments suivants :

```
{
    "Name": "ExampleFLEProfileUpdated",
    "CallerReference": "cli-example",
    "Comment": "Updated FLE profile for AWS CLI example",
    "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
            {
                "PublicKeyId": "K2K8NC4HVFE3M0",
                "ProviderId": "ExampleFLEProvider",
                "FieldPatterns": {
                    "Quantity": 2,
                    "Items": [
                        "ExampleSensitiveField",
```

```
                    "SecondExampleSensitiveField"
                ]
            }
        }
    ]
}
}
```

Sortie :

```
{
    "ETag": "EJETYFJ9CL66D",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T19:05:58.296Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfileUpdated",
            "CallerReference": "cli-example",
            "Comment": "Updated FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 2,
                            "Items": [
                                "ExampleSensitiveField",
                                "SecondExampleSensitiveField"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateFieldLevelEncryptionProfileà la section Référence des AWS CLI commandes.

`verify-dns-configuration`

L'exemple de code suivant montre comment utiliser`verify-dns-configuration`.

AWS CLI

Pour vérifier la configuration DNS d'un domaine

L'`verify-dns-configuration`exemple suivant vérifie la configuration DNS d'un domaine.

```
aws cloudfront verify-dns-configuration \
    --domain example.com \
    --identifier dt_2x9GhoK0TZRsohWzv1b9It8J1AB
```

Sortie :

```
{
    "DnsConfigurationList": [
        {
            "Domain": "example.com",
            "Status": "valid-configuration"
        }
    ]
}
```

Pour plus d'informations, consultez la section Déplacer un autre nom de domaine vers une autre distribution dans le manuel Amazon CloudFront Developer Guide.

- Pour plus de détails sur l'API, reportez-vous VerifyDnsConfigurationà la section Référence des AWS CLI commandes.

CloudSearch Exemples d'Amazon utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon CloudSearch.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`upload-documents`

L'exemple de code suivant montre comment utiliser`upload-documents`.

AWS CLI

La `upload-documents` commande suivante télécharge un lot de documents JSON vers un CloudSearch domaine Amazon :

```
aws cloudsearchdomain upload-documents --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com --content-type application/json --documents document-batch.json
```

Sortie :

```
{
  "status": "success",
  "adds": 5000,
  "deletes": 0
}
```

- Pour plus de détails sur l'API, reportez-vous UploadDocumentsà la section Référence des AWS CLI commandes.

CloudTrail exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CloudTrail.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags`

L'exemple de code suivant montre comment utiliser`add-tags`.

AWS CLI

Pour ajouter des balises au parcours

La `add-tags` commande suivante ajoute des balises pour `Trail1` :

```
aws cloudtrail add-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-
list Key=name,Value=Alice Key=location,Value=us
```

- Pour plus de détails sur l'API, reportez-vous AddTagsà la section Référence des AWS CLI commandes.

`create-subscription`

L'exemple de code suivant montre comment utiliser`create-subscription`.

AWS CLI

Pour créer et configurer AWS des ressources pour un parcours

La `create-subscription` commande suivante crée un nouveau compartiment S3 et une nouvelle rubrique SNS pour`Trail1`.

```
aws cloudtrail create-subscription \
    --name Trail1 \
    --s3-new-bucket amzn-s3-demo-bucket \
    --sns-new-topic my-topic
```

Sortie :

```
Setting up new S3 bucket amzn-s3-demo-bucket...
Setting up new SNS topic my-topic...
Creating/updating CloudTrail configuration...
CloudTrail configuration:
    {
        "trailList": [
            {
                "IncludeGlobalServiceEvents": true,
                "Name": "Trail1",
                "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/
Trail1",
                "LogFileValidationEnabled": false,
                "IsMultiRegionTrail": false,
                "S3BucketName": "amzn-s3-demo-bucket",
                "SnsTopicName": "my-topic",
                "HomeRegion": "us-east-1"
            }
        ],
        "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "RequestId": "f39e51f6-c615-11e5-85bd-d35ca21ee3e2"
        }
    }
Starting CloudTrail service...
Logs will be delivered to my-bucket
```

- Pour plus de détails sur l'API, reportez-vous CreateSubscriptionà la section Référence des AWS CLI commandes.

`create-trail`

L'exemple de code suivant montre comment utiliser`create-trail`.

AWS CLI

Pour créer un parcours

L'`create-trail`exemple suivant crée un journal multirégional nommé `Trail1` et spécifie un compartiment S3.

```
aws cloudtrail create-trail \
    --name Trail1 \
```

```
    --s3-bucket-name amzn-s3-demo-bucket \
    --is-multi-region-trail
```

Sortie :

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Trail1",
    "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "S3BucketName": "amzn-s3-demo-bucket"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateTrailà la section Référence des AWS CLI commandes.

`delete-trail`

L'exemple de code suivant montre comment utiliser`delete-trail`.

AWS CLI

Pour supprimer un parcours

La `delete-trail` commande suivante supprime une piste nommée `Trail1` :

```
aws cloudtrail delete-trail --name Trail1
```

- Pour plus de détails sur l'API, reportez-vous DeleteTrailà la section Référence des AWS CLI commandes.

`describe-trails`

L'exemple de code suivant montre comment utiliser`describe-trails`.

AWS CLI

Pour décrire un sentier

L'`describe-trails`exemple suivant renvoie les paramètres pour `Trail1` et`Trail2`.

```
aws cloudtrail describe-trails \
    --trail-name-list Trail1 Trail2
```

Sortie :

```
{
    "trailList": [
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Trail1",
            "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
            "LogFileValidationEnabled": false,
            "IsMultiRegionTrail": false,
            "S3BucketName": "amzn-s3-demo-bucket",
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/
CloudTrail_CloudWatchLogs_Role",
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-
group:CloudTrail:*",
            "SnsTopicName": "my-topic",
            "HomeRegion": "us-east-1"
        },
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Trail2",
            "S3KeyPrefix": "my-prefix",
            "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
            "LogFileValidationEnabled": false,
            "IsMultiRegionTrail": false,
            "S3BucketName": "amzn-s3-demo-bucket2",
            "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/4c5ae5ac-3c13-421e-8335-c7868ef6a769",
            "HomeRegion": "us-east-1"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTrailsà la section Référence des AWS CLI commandes.

`get-event-selectors`

L'exemple de code suivant montre comment utiliser`get-event-selectors`.

AWS CLI

Pour consulter les paramètres du sélecteur d'événements pour un parcours

La `get-event-selectors` commande suivante renvoie les paramètres pour `Trail1` :

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

Sortie :

```
{
  "EventSelectors": [
      {
          "IncludeManagementEvents": true,
          "DataResources": [],
          "ReadWriteType": "All"
      }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
}
```

- Pour plus de détails sur l'API, reportez-vous GetEventSelectorsà la section Référence des AWS CLI commandes.

`get-trail-status`

L'exemple de code suivant montre comment utiliser`get-trail-status`.

AWS CLI

Pour obtenir le statut d'une piste

La `get-trail-status` commande suivante renvoie les informations de livraison et de journalisation pour `Trail1` :

```
aws cloudtrail get-trail-status --name Trail1
```

Sortie :

```
{
```

```
  "LatestNotificationTime": 1454022144.869,
  "LatestNotificationAttemptSucceeded": "2016-01-28T23:02:24Z",
  "LatestDeliveryAttemptTime": "2016-01-28T23:02:24Z",
  "LatestDeliveryTime": 1454022144.869,
  "TimeLoggingStarted": "2015-11-06T18:36:38Z",
  "LatestDeliveryAttemptSucceeded": "2016-01-28T23:02:24Z",
  "IsLogging": true,
  "LatestCloudWatchLogsDeliveryTime": 1454022144.918,
  "StartLoggingTime": 1446834998.695,
  "StopLoggingTime": 1446834996.933,
  "LatestNotificationAttemptTime": "2016-01-28T23:02:24Z",
  "TimeLoggingStopped": "2015-11-06T18:36:36Z"
}
```

- Pour plus de détails sur l'API, reportez-vous GetTrailStatusà la section Référence des AWS CLI commandes.

`list-public-keys`

L'exemple de code suivant montre comment utiliser`list-public-keys`.

AWS CLI

Pour répertorier toutes les clés publiques d'un parcours

La `list-public-keys` commande suivante renvoie toutes les clés publiques dont les clés privées ont été utilisées pour signer les fichiers de résumé dans le délai spécifié :

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

Sortie :

```
{
  "PublicKeyList": [
      {
         "ValidityStartTime": 1453076702.0,
         "ValidityEndTime": 1455668702.0,
         "Value": "MIIBCgKCAQEAlSS3cl92HDycr/MTj0moOhas8habjrraXw+KzlWF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kUl1xboEcEPHYoIYMbgfSw7KGnuDKwkLzsQWhUJ0cIbOHASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmUl8/P7XpfpWQuk4h8g3pXZOrNXr08lbh4d39svj7UqdhvOXoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",
```

```
            "Fingerprint": "7f3f401420072e50a65a141430817ab3"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPublicKeysà la section Référence des AWS CLI commandes.

`list-tags`

L'exemple de code suivant montre comment utiliser`list-tags`.

AWS CLI

Pour répertorier les balises d'un parcours

La `list-tags` commande suivante répertorie les balises pour `Trail1` et `Trail2` :

```
aws cloudtrail list-tags --resource-id-list arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 arn:aws:cloudtrail:us-east-1:123456789012:trail/
Trail2
```

Sortie :

```
{
 "ResourceTagList": [
     {
         "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
         "TagsList": [
             {
                 "Value": "Alice",
                 "Key": "name"
             },
             {
                 "Value": "us",
                 "Key": "location"
             }
         ]
     },
     {
         "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
```

```
            "TagsList": [
                {
                    "Value": "Bob",
                    "Key": "name"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsà la section Référence des AWS CLI commandes.

`lookup-events`

L'exemple de code suivant montre comment utiliser`lookup-events`.

AWS CLI

Pour rechercher des événements pour un parcours

La `lookup-events` commande suivante recherche les événements d'activité de l'API par attribut `EventName` :

```
aws cloudtrail lookup-events --lookup-
attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

Sortie :

```
{
  "Events": [
     {
         "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
         "Username": "my-session-name",
         "EventTime": "2021-11-18T09:41:02-08:00",
         "CloudTrailEvent": "{\"eventVersion\":\"1.02\",\"userIdentity\":{\"type\":
\"AssumedRole\",\"principalId\":\"AROAJIKPFTA72SWU4L7T4:my-session-name\",\"arn\":
\"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\",\"accountId\":
\"123456789012\",\"sessionContext\":{\"attributes\":{\"mfaAuthenticated\":\"false
\",\"creationDate\":\"2016-01-26T21:42:12Z\"},\"sessionIssuer\":{\"type\":\"Role\",
\"principalId\":\"AROAJIKPFTA72SWU4L7T4\",\"arn\":\"arn:aws:iam::123456789012:role/
```

```
my-role\",\"accountId\":\"123456789012\",\"userName\":\"my-role\"}}},\"eventTime
\":\"2016-01-26T21:42:12Z\",\"eventSource\":\"signin.amazonaws.com\",\"eventName\":
\"ConsoleLogin\",\"awsRegion\":\"us-east-1\",\"sourceIPAddress\":\"72.21.198.70\",
\"userAgent\":\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36\",\"requestParameters
\":null,\"responseElements\":{\"ConsoleLogin\":\"Success\"},\"additionalEventData\":
{\"MobileVersion\":\"No\",\"MFAUsed\":\"No\"},\"eventID\":\"654ccbc0-ba0d-486a-9076-
dbf7274677a7\",\"eventType\":\"AwsConsoleSignIn\",\"recipientAccountId\":
\"123456789012\"}",
        "EventName": "ConsoleLogin",
        "Resources": []
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous LookupEventsà la section Référence des AWS CLI commandes.

`put-event-selectors`

L'exemple de code suivant montre comment utiliser`put-event-selectors`.

AWS CLI

Exemple 1 : configurer un journal pour consigner les événements de gestion et les événements de données à l'aide de sélecteurs d'événements avancés

Vous pouvez ajouter des sélecteurs d'événements avancés et des conditions pour vos sélecteurs d'événements avancés, jusqu'à un maximum de 500 valeurs pour toutes les conditions et tous les sélecteurs d'un sentier. Vous pouvez utiliser des sélecteurs d'événements avancés pour enregistrer tous les types d'événements de données disponibles. Vous pouvez utiliser des sélecteurs d'événements avancés ou des sélecteurs d'événements de base, mais pas les deux. Si vous appliquez des sélecteurs d’événements avancés à un journal de suivi, tous les sélecteurs d’événements de base existants sont remplacés.

L'`put-event-selectors`exemple suivant crée un sélecteur d'événements avancé pour un journal nommé `myTrail` afin de consigner tous les événements de gestion, de consigner les appels S3 PutObject et d' DeleteObject API pour tous les compartiments S3 sauf un, de consigner les appels d'API de données pour une fonction Lambda `myFunction` nommée et de consigner les appels d'API de publication sur un sujet SNS nommé. `myTopic`

```
aws cloudtrail put-event-selectors \
    --trail-name myTrail \
    --advanced-event-selectors '[{"Name": "Log all management events",
 "FieldSelectors": [{ "Field": "eventCategory", "Equals": ["Management"] }] },
{"Name": "Log PutObject and DeleteObject events for all but one
 bucket","FieldSelectors": [{ "Field": "eventCategory", "Equals": ["Data"] },
{ "Field": "resources.type", "Equals": ["AWS::S3::Object"] },{ "Field":
 "eventName", "Equals": ["PutObject","DeleteObject"] },{ "Field": "resources.ARN",
 "NotStartsWith": ["arn:aws:s3:::amzn-s3-demo-bucket/"] }]},{"Name": "Log
 data events for a specific Lambda function","FieldSelectors": [{ "Field":
 "eventCategory", "Equals": ["Data"] },{ "Field": "resources.type",
 "Equals": ["AWS::Lambda::Function"] },{ "Field": "resources.ARN", "Equals":
 ["arn:aws:lambda:us-east-1:123456789012:function:myFunction"] }]},{"Name":
 "Log all Publish API calls on a specific SNS topic","FieldSelectors":
 [{ "Field": "eventCategory", "Equals": ["Data"] },{ "Field": "resources.type",
 "Equals": ["AWS::SNS::Topic"] },{ "Field": "eventName", "Equals":
 ["Publish"] },{ "Field": "resources.ARN", "Equals": ["arn:aws:sns:us-
east-1:123456789012:myTopic.fifo"] }]}]'
```

Sortie :

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/myTrail",
    "AdvancedEventSelectors": [
        {
            "Name": "Log all management events",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Management"
                    ]
                }
            ]
        },
        {
            "Name": "Log PutObject and DeleteObject events for all but one bucket",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
```

```
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                },
                {
                    "Field": "eventName",
                    "Equals": [
                        "PutObject",
                        "DeleteObject"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "NotStartsWith": [
                        "arn:aws:s3:::amzn-s3-demo-bucket/"
                    ]
                }
            ]
        },
        {
            "Name": "Log data events for a specific Lambda function",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Lambda::Function"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "Equals": [
                        "arn:aws:lambda:us-east-1:123456789012:function:myFunction"
                    ]
                }
            ]
```

```
        },
        {
            "Name": "Log all Publish API calls on a specific SNS topic",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SNS::Topic"
                    ]
                },
                {
                    "Field": "eventName",
                    "Equals": [
                        "Publish"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "Equals": [
                        "arn:aws:sns:us-east-1:123456789012:myTopic.fifo"
                    ]
                }
            ]
        }
    ]
}
```

Pour plus d'informations, voir Enregistrer les événements à l'aide de sélecteurs d'événements avancés dans le Guide de l'AWS CloudTrail utilisateur.

Exemple 2 : configurer les sélecteurs d'événements pour un journal afin de consigner tous les événements de gestion et les événements de données

Vous pouvez configurer jusqu'à 5 sélecteurs d'événements et jusqu'à 250 ressources de données pour un journal de suivi. Les sélecteurs d'événements sont également appelés sélecteurs d'événements de base. Vous pouvez utiliser des sélecteurs d'événements pour consigner les événements de gestion et les événements de données pour les objets S3, les fonctions Lambda

et les tables DynnaMoDB. Pour enregistrer des événements de données pour d'autres types de ressources, vous devez utiliser des sélecteurs d'événements avancés.

L'`put-event-selectors`exemple suivant crée un sélecteur d'événements pour un journal nommé de manière `TrailName` à inclure tous les événements de gestion, les événements de données pour deux bucket/prefix combinaisons Amazon S3 et les événements de données pour une seule fonction AWS Lambda nommée. `hello-world-python-function`

```
aws cloudtrail put-event-selectors \
    --trail-name TrailName \
    --event-selectors '[{"ReadWriteType": "All","IncludeManagementEvents":
 true,"DataResources": [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3:::amzn-
s3-demo-bucket/prefix","arn:aws:s3:::amzn-s3-demo-bucket2/prefix2"]},
{"Type": "AWS::Lambda::Function","Values": ["arn:aws:lambda:us-
west-2:999999999999:function:hello-world-python-function"]}]}]'
```

Sortie :

```
{
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
                        "arn:aws:s3:::amzn-s3-demo-bucket/prefix",
                        "arn:aws:s3:::amzn-s3-demo-bucket2/prefix2"
                    ],
                    "Type": "AWS::S3::Object"
                },
                {
                    "Values": [
                        "arn:aws:lambda:us-west-2:123456789012:function:hello-world-
python-function"
                    ],
                    "Type": "AWS::Lambda::Function"
                },
            ],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
```

```
}
```

Pour plus d'informations, consultez la section Consigner les événements à l'aide des sélecteurs d'événements de base du Guide de l'AWS CloudTrail utilisateur.

Exemple 3 : configurer des sélecteurs d'événements pour un journal afin de consigner les événements de gestion, tous les événements de données S3 sur les objets S3 et tous les événements de données Lambda sur les fonctions de votre compte

L'`put-event-selectors`exemple suivant crée un sélecteur d'événements pour un journal nommé `TrailName2` qui inclut tous les événements de gestion et tous les événements de données pour tous les buckets Amazon S3 et les fonctions AWS Lambda du compte. AWS

```
aws cloudtrail put-event-selectors \
    --trail-name TrailName2 \
    --event-selectors '[{"ReadWriteType": "All","IncludeManagementEvents":
 true,"DataResources": [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3"]},
{"Type": "AWS::Lambda::Function","Values": ["arn:aws:lambda"]}]}]'
```

Sortie :

```
{
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
                        "arn:aws:s3"
                    ],
                    "Type": "AWS::S3::Object"
                },
                {
                    "Values": [
                        "arn:aws:lambda"
                    ],
                    "Type": "AWS::Lambda::Function"
                },
            ],
            "ReadWriteType": "All"
        }
```

```
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName2"
}
```

Pour plus d'informations, consultez la section Consigner les événements à l'aide des sélecteurs d'événements de base du Guide de l'AWS CloudTrail utilisateur.

- Pour plus de détails sur l'API, reportez-vous PutEventSelectorsà la section Référence des AWS CLI commandes.

`remove-tags`

L'exemple de code suivant montre comment utiliser`remove-tags`.

AWS CLI

Pour supprimer les balises d'un parcours

La `remove-tags` commande suivante supprime les balises spécifiées pour `Trail1` :

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- Pour plus de détails sur l'API, reportez-vous RemoveTagsà la section Référence des AWS CLI commandes.

`start-logging`

L'exemple de code suivant montre comment utiliser`start-logging`.

AWS CLI

Pour commencer à enregistrer un parcours

La `start-logging` commande suivante active la journalisation pour `Trail1` :

```
aws cloudtrail start-logging --name Trail1
```

- Pour plus de détails sur l'API, reportez-vous StartLoggingà la section Référence des AWS CLI commandes.

`stop-logging`

L'exemple de code suivant montre comment utiliser`stop-logging`.

AWS CLI

Pour arrêter d'enregistrer un parcours

La `stop-logging` commande suivante désactive la journalisation pour `Trail1` :

```
aws cloudtrail stop-logging --name Trail1
```

- Pour plus de détails sur l'API, reportez-vous StopLoggingà la section Référence des AWS CLI commandes.

`update-subscription`

L'exemple de code suivant montre comment utiliser`update-subscription`.

AWS CLI

Pour mettre à jour les paramètres de configuration d'un parcours

L'`update-subscription`exemple suivant met à jour le journal pour spécifier un nouveau compartiment S3 et une nouvelle rubrique SNS.

```
aws cloudtrail update-subscription \
    --name Trail1 \
    --s3-new-bucket amzn-s3-demo-bucket \
    --sns-new-topic my-topic-new
```

Sortie :

```
Setting up new S3 bucket amzn-s3-demo-bucket...
Setting up new SNS topic my-topic-new...
Creating/updating CloudTrail configuration...
CloudTrail configuration:
{
    "trailList": [
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Trail1",
```

```
            "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
            "LogFileValidationEnabled": false,
            "IsMultiRegionTrail": false,
            "S3BucketName": "amzn-s3-demo-bucket",
            "SnsTopicName": "my-topic-new",
            "HomeRegion": "us-east-1"
        }
    ],
    "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "RequestId": "31126f8a-c616-11e5-9cc6-2fd637936879"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateSubscriptionà la section Référence des AWS CLI commandes.

`update-trail`

L'exemple de code suivant montre comment utiliser`update-trail`.

AWS CLI

Pour mettre à jour un parcours

L'`update-trail`exemple suivant met à jour un journal afin d'utiliser un bucket existant pour la livraison du journal.

```
aws cloudtrail update-trail \
    --name Trail1 \
    --s3-bucket-name amzn-s3-demo-bucket
```

Sortie :

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Trail1",
    "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "S3BucketName": "amzn-s3-demo-bucket"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateTrailà la section Référence des AWS CLI commandes.

`validate-logs`

L'exemple de code suivant montre comment utiliser`validate-logs`.

AWS CLI

Pour valider un fichier journal

La `validate-logs` commande suivante valide les journaux pour `Trail1` :

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

Sortie :

```
Validating log files for trail arn:aws:cloudtrail:us-east-1:123456789012:trail/
Trail1 between 2016-01-29T19:00:00Z and 2016-01-29T22:15:43Z
Results requested for 2016-01-29T19:00:00Z to 2016-01-29T22:15:43Z
Results found for 2016-01-29T19:24:57Z to 2016-01-29T21:24:57Z:
3/3 digest files valid
15/15 log files valid
```

- Pour plus de détails sur l'API, reportez-vous ValidateLogsà la section Référence des AWS CLI commandes.

CloudWatch exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CloudWatch.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-alarms`

L'exemple de code suivant montre comment utiliser`delete-alarms`.

AWS CLI

Pour supprimer une alarme

L'exemple suivant utilise la `delete-alarms` commande pour supprimer l' CloudWatch alarme Amazon nommée « myalarm » :

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Sortie :

```
This command returns to the prompt if successful.
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà la section Référence des AWS CLI commandes.

`delete-anomaly-detector`

L'exemple de code suivant montre comment utiliser`delete-anomaly-detector`.

AWS CLI

Pour supprimer un modèle de détection d'anomalie spécifié

L'`delete-anomaly-detector`exemple suivant supprime un modèle de détecteur d'anomalies dans le compte spécifié.

```
aws cloudwatch delete-anomaly-detector \
    --namespace AWS/Logs \
    --metric-name IncomingBytes \
    --stat SampleCount
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un modèle de détection d'anomalies dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteAnomalyDetectorà la section Référence des AWS CLI commandes.

`delete-dashboards`

L'exemple de code suivant montre comment utiliser`delete-dashboards`.

AWS CLI

Pour supprimer des tableaux de bord spécifiés

L'`delete-dashboards`exemple suivant supprime deux tableaux de bord nommés `Dashboard-A` et `Dashboard-B` dans le compte spécifié.

```
aws cloudwatch delete-dashboards \
    --dashboard-names Dashboard-A Dashboard-B
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les CloudWatch tableaux de bord Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteDashboardsà la section Référence des AWS CLI commandes.

`delete-insight-rules`

L'exemple de code suivant montre comment utiliser`delete-insight-rules`.

AWS CLI

Pour supprimer des règles spécifiques relatives aux informations sur les contributeurs

L'`delete-insight-rules`exemple suivant supprime deux règles d'analyse des contributeurs nommées `Rule-A` et `Rule-B` dans le compte spécifié.

```
aws cloudwatch delete-insight-rules \
    --rule-names Rule-A Rule-B
```

Sortie :

```
{
    "Failures": []
}
```

Pour plus d'informations, consultez la section Utiliser Contributor Insights pour analyser les données de cardinalité élevée dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteInsightRulesà la section Référence des AWS CLI commandes.

`delete-metric-stream`

L'exemple de code suivant montre comment utiliser`delete-metric-stream`.

AWS CLI

Pour supprimer un flux métrique spécifié

L'`delete-metric-stream`exemple suivant supprime le flux métrique nommé `QuickPartial-gSCKv0` dans le compte spécifié.

```
aws cloudwatch delete-metric-stream \
    --name QuickPartial-gSCKv0
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les flux métriques dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteMetricStreamà la section Référence des AWS CLI commandes.

`describe-alarm-history`

L'exemple de code suivant montre comment utiliser`describe-alarm-history`.

AWS CLI

Pour récupérer l'historique d'une alarme

L'exemple suivant utilise la `describe-alarm-history` commande pour récupérer l'historique de l'CloudWatch alarme Amazon nommée « myalarm » :

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-
type StateUpdate
```

Sortie :

```
{
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue\":
\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue\":\"OK
\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not greater than the
 threshold (70.0). The most recent datapoints: [38.958, 40.292].\",\"stateReasonData
\":{\"version\":\"1.0\",\"queryDate\":\"2014-04-09T18:59:06.419+0000\",\"startDate
\":\"2014-04-09T18:44:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.958,40.292],\"threshold\":70.0}}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState\":
{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmHistoryà la section Référence des AWS CLI commandes.

`describe-alarms-for-metric`

L'exemple de code suivant montre comment utiliser`describe-alarms-for-metric`.

AWS CLI

Pour afficher des informations relatives aux alarmes associées à une métrique

L'exemple suivant utilise la `describe-alarms-for-metric` commande pour afficher des informations sur les alarmes associées à la EC2 CPUUtilization métrique Amazon et à l'instance portant l'ID i-0c986c72. :

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Sortie :

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
```

```
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not greater than
 or equal to the threshold (70.0). The most recent datapoints: [40.760000000000005,
 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": false,
```

```
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà la section Référence des AWS CLI commandes.

`describe-alarms`

L'exemple de code suivant montre comment utiliser`describe-alarms`.

AWS CLI

Pour répertorier les informations relatives à une alarme

L'exemple suivant fait appel à la commande `describe-alarms` pour fournir des informations sur l'alarme nommée « myalarm » :

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Sortie :

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
```

```
            "Threshold": 70.0,
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà la section Référence des AWS CLI commandes.

`describe-anomaly-detectors`

L'exemple de code suivant montre comment utiliser`describe-anomaly-detectors`.

AWS CLI

Pour récupérer une liste de modèles de détection d'anomalies

L'`describe-anomaly-detectors`exemple suivant affiche des informations sur les modèles de détecteurs d'anomalies associés à l'espace de `AWS/Logs` noms du compte spécifié.

```
aws cloudwatch describe-anomaly-detectors \
    --namespace AWS/Logs
```

Sortie :

```
{
    "AnomalyDetectors": [
        {
            "Namespace": "AWS/Logs",
```

```
            "MetricName": "IncomingBytes",
            "Dimensions": [],
            "Stat": "SampleCount",
            "Configuration": {
                "ExcludedTimeRanges": []
            },
            "StateValue": "TRAINED",
            "SingleMetricAnomalyDetector": {
                "AccountId": "123456789012",
                "Namespace": "AWS/Logs",
                "MetricName": "IncomingBytes",
                "Dimensions": [],
                "Stat": "SampleCount"
            }
        },
        {
            "Namespace": "AWS/Logs",
            "MetricName": "IncomingBytes",
            "Dimensions": [
                {
                    "Name": "LogGroupName",
                    "Value": "demo"
                }
            ],
            "Stat": "Average",
            "Configuration": {
                "ExcludedTimeRanges": []
            },
            "StateValue": "PENDING_TRAINING",
            "SingleMetricAnomalyDetector": {
                "AccountId": "123456789012",
                "Namespace": "AWS/Logs",
                "MetricName": "IncomingBytes",
                "Dimensions": [
                    {
                        "Name": "LogGroupName",
                        "Value": "demo"
                    }
                ],
                "Stat": "Average"
            }
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Utilisation de la détection des CloudWatch anomalies dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeAnomalyDetectorsà la section Référence des AWS CLI commandes.

`describe-insight-rules`

L'exemple de code suivant montre comment utiliser`describe-insight-rules`.

AWS CLI

Pour récupérer une liste des règles de Contributor Insights

L'`describe-insight-rules`exemple suivant montre toutes les règles de Contributor Insight dans le compte spécifié.

```
aws cloudwatch describe-insight-rules
```

Sortie :

```
{
    "InsightRules": [
        {
            "Name": "Rule-A",
            "State": "ENABLED",
            "Schema": "CloudWatchLogRule/1",
            "Definition": "{\n\t\"AggregateOn\": \"Count\",\n\t\"Contribution\":
 {\n\t\t\"Filters\": [],\n\t\t\"Keys\": [\n\t\t\t\"$.requestId\"\n\t\t]\n\t},\n
\t\"LogFormat\": \"JSON\",\n\t\"Schema\": {\n\t\t\"Name\": \"CloudWatchLogRule
\",\n\t\t\"Version\": 1\n\t},\n\t\"LogGroupARNs\": [\n\t\t\"arn:aws:logs:us-
east-1:123456789012:log-group:demo\"\n\t]\n}",
            "ManagedRule": false
        },
        {
            "Name": "Rule-B",
            "State": "ENABLED",
            "Schema": "CloudWatchLogRule/1",
            "Definition": "{\n\t\"AggregateOn\": \"Count\",\n\t\"Contribution\":
 {\n\t\t\"Filters\": [],\n\t\t\"Keys\": [\n\t\t\t\"$.requestId\"\n\t\t]\n\t},\n
```

```
\t\"LogFormat\": \"JSON\",\n\t\"Schema\": {\n\t\t\"Name\": \"CloudWatchLogRule
\",\n\t\t\"Version\": 1\n\t},\n\t\"LogGroupARNs\": [\n\t\t\"arn:aws:logs:us-
east-1:123456789012:log-group:demo-1\"\n\t]\n}",
            "ManagedRule": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Utiliser Contributor Insights pour analyser les données de cardinalité élevée dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInsightRulesà la section Référence des AWS CLI commandes.

`disable-alarm-actions`

L'exemple de code suivant montre comment utiliser`disable-alarm-actions`.

AWS CLI

Pour désactiver les actions d’une alarme

L’exemple suivant fait appel à la commande `disable-alarm-actions` pour désactiver toutes les actions de l’alarme nommée myalarm :

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Cette commande revient à l’invite en cas de succès.

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section Référence des AWS CLI commandes.

`disable-insight-rules`

L'exemple de code suivant montre comment utiliser`disable-insight-rules`.

AWS CLI

Pour désactiver les règles d'analyse des contributeurs spécifiées

L'`disable-insight-rules`exemple suivant désactive deux règles d'analyse des contributeurs nommées `Rule-A` et `Rule-B` dans le compte spécifié.

```
aws cloudwatch disable-insight-rules \
    --rule-names Rule-A Rule-B
```

Sortie :

```
{
    "Failures": []
}
```

Pour plus d'informations, consultez la section Utiliser Contributor Insights pour analyser les données de cardinalité élevée dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisableInsightRulesà la section Référence des AWS CLI commandes.

`enable-alarm-actions`

L'exemple de code suivant montre comment utiliser`enable-alarm-actions`.

AWS CLI

Pour activer toutes les actions d’une alarme

L’exemple suivant fait appel à la commande `enable-alarm-actions` pour activer toutes les actions de l’alarme nommée myalarm :

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Cette commande revient à l’invite en cas de succès.

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà la section Référence des AWS CLI commandes.

`enable-insight-rules`

L'exemple de code suivant montre comment utiliser`enable-insight-rules`.

AWS CLI

Pour activer les règles d'analyse des contributeurs spécifiées

L'`enable-insight-rules`exemple suivant active deux règles d'analyse des contributeurs nommées `Rule-A` et `Rule-B` dans le compte spécifié.

```
aws cloudwatch enable-insight-rules \
    --rule-names Rule-A Rule-B
```

Sortie :

```
{
    "Failures": []
}
```

Pour plus d'informations, consultez la section Utiliser Contributor Insights pour analyser les données de cardinalité élevée dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous EnableInsightRulesà la section Référence des AWS CLI commandes.

`get-dashboard`

L'exemple de code suivant montre comment utiliser`get-dashboard`.

AWS CLI

Pour récupérer des informations sur un tableau de bord

L'`get-dashboard`exemple suivant affiche des informations sur le tableau de bord nommé `Dashboard-A` dans le compte spécifié.

```
aws cloudwatch get-dashboard \
    --dashboard-name Dashboard-A
```

Sortie :

```
{
    "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-A",
    "DashboardBody": "{\"widgets\":[{\"type\":\"metric\",\"x\":0,\"y\":0,\"width
\":6,\"height\":6,\"properties\":{\"view\":\"timeSeries\",\"stacked\":false,
\"metrics\":[[\"AWS/EC2\",\"NetworkIn\",\"InstanceId\",\"i-0131f062232ade043\"],[\".
\",\"NetworkOut\",\".\",\".\"]],\"region\":\"us-east-1\"}}]}",
    "DashboardName": "Dashboard-A"
```

```
}
```

Pour plus d'informations, consultez les CloudWatch tableaux de bord Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetDashboardà la section Référence des AWS CLI commandes.

`get-insight-rule-report`

L'exemple de code suivant montre comment utiliser`get-insight-rule-report`.

AWS CLI

Pour récupérer les données de séries chronologiques collectées par une règle Contributor Insights

L'`get-insight-rule-report`exemple suivant renvoie les données de séries chronologiques collectées par une règle Contributor Insights.

```
aws cloudwatch get-insight-rule-report \
    --rule-name Rule-A \
    --start-time 2024-10-13T20:15:00Z \
    --end-time 2024-10-13T20:30:00Z \
    --period 300
```

Sortie :

```
{
    "KeyLabels": [
        "PartitionKey"
    ],
    "AggregationStatistic": "Sum",
    "AggregateValue": 0.5,
    "ApproximateUniqueCount": 1,
    "Contributors": [
        {
            "Keys": [
                "RequestID"
            ],
            "ApproximateAggregateValue": 0.5,
            "Datapoints": [
```

```
                {
                    "Timestamp": "2024-10-13T21:00:00+00:00",
                    "ApproximateValue": 0.5
                }
            ]
        }
    ],
    "RuleAttributes": []
}
```

Pour plus d'informations, consultez la section Utiliser Contributor Insights pour analyser les données de cardinalité élevée dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetInsightRuleReportà la section Référence des AWS CLI commandes.

`get-metric-data`

L'exemple de code suivant montre comment utiliser`get-metric-data`.

AWS CLI

Exemple 1 : Pour obtenir le total moyen d'IOPS pour le paramètre spécifié à l' EC2 aide d'une expression mathématique

L'`get-metric-data`exemple suivant récupère les valeurs CloudWatch métriques de l' EC2 instance associée à `i-abcdef` InstanceID à l'aide d'une expression mathématique qui combine des métriques et des métriques. `EBSReadOps` `EBSWriteOps`

```
aws cloudwatch get-metric-data \
    --metric-data-queries file://file.json \
    --start-time 2024-09-29T22:10:00Z \
    --end-time 2024-09-29T22:15:00Z
```

Contenu de `file.json` :

```
[
    {
        "Id": "m3",
        "Expression": "(m1+m2)/300",
        "Label": "Avg Total IOPS"
    },
```

```
    {
        "Id": "m1",
        "MetricStat": {
            "Metric": {
                "Namespace": "AWS/EC2",
                "MetricName": "EBSReadOps",
                "Dimensions": [
                    {
                        "Name": "InstanceId",
                        "Value": "i-abcdef"
                    }
                ]
            },
            "Period": 300,
            "Stat": "Sum",
            "Unit": "Count"
        },
        "ReturnData": false
    },
    {
        "Id": "m2",
        "MetricStat": {
            "Metric": {
                "Namespace": "AWS/EC2",
                "MetricName": "EBSWriteOps",
                "Dimensions": [
                    {
                        "Name": "InstanceId",
                        "Value": "i-abcdef"
                    }
                ]
            },
            "Period": 300,
            "Stat": "Sum",
            "Unit": "Count"
        },
        "ReturnData": false
    }
]
```

Sortie :

```
{
```

```
    "MetricDataResults": [
        {
            "Id": "m3",
            "Label": "Avg Total IOPS",
            "Timestamps": [
                "2024-09-29T22:10:00+00:00"
            ],
            "Values": [
                96.85
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Exemple 2 : pour surveiller les AWS frais estimés à l'aide des indicateurs CloudWatch de facturation

L'`get-metric-data`exemple suivant extrait la `EstimatedCharges` CloudWatch métrique de l'espace de noms AWS/Billing.

```
aws cloudwatch get-metric-data \
    --metric-data-queries '[{"Id":"m1","MetricStat":{"Metric":
{"Namespace":"AWS/Billing","MetricName":"EstimatedCharges","Dimensions":
[{"Name":"Currency","Value":"USD"}]},"Period":21600,"Stat":"Maximum"}}]' \
    --start-time 2024-09-26T12:00:00Z \
    --end-time 2024-09-26T18:00:00Z \
    --region us-east-1
```

Sortie :

```
{
    "MetricDataResults": [
        {
            "Id": "m1",
            "Label": "EstimatedCharges",
            "Timestamps": [
                "2024-09-26T12:00:00+00:00"
            ],
            "Values": [
                542.38
            ],
```

```
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Pour plus d'informations, consultez la section Utilisation d'expressions mathématiques avec CloudWatch des métriques dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetMetricDataà la section Référence des AWS CLI commandes.

`get-metric-statistics`

L'exemple de code suivant montre comment utiliser`get-metric-statistics`.

AWS CLI

Pour obtenir l'utilisation du processeur par EC2 instance

L'exemple suivant utilise la `get-metric-statistics` commande pour obtenir l'utilisation du processeur pour une EC2 instance portant l'ID i-abcdef.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-
time 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --
namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Sortie :

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
```

```
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
```

```
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
```

```
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

Spécification de plusieurs dimensions

L'exemple suivant illustre comment spécifier plusieurs dimensions. Chaque dimension est spécifiée sous forme de Name/Value paire, avec une virgule entre le nom et la valeur. Les différentes dimensions sont séparées par une espace. Si une métrique comprend plusieurs dimensions, vous devez préciser une valeur pour chaque dimension définie.

Pour d'autres exemples d'utilisation de cette `get-metric-statistics` commande, consultez la section Obtenir des statistiques pour une métrique dans le manuel Amazon CloudWatch Developer Guide.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace
 --dimensions Name=InstanceID,Value=i-abcdef Name=InstanceType,Value=m1.small --
start-time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average
 --period 60
```

- Pour plus de détails sur l'API, reportez-vous GetMetricStatisticsà la section Référence des AWS CLI commandes.

`get-metric-stream`

L'exemple de code suivant montre comment utiliser`get-metric-stream`.

AWS CLI

Pour récupérer des informations sur un flux métrique

L'`get-metric-stream`exemple suivant affiche des informations sur le flux métrique nommé `QuickFull-GuaFbs` dans le compte spécifié.

```
aws cloudwatch get-metric-stream \
    --name QuickFull-GuaFbs
```

Sortie :

```
{
    "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/QuickFull-
GuaFbs",
    "Name": "QuickFull-GuaFbs",
    "FirehoseArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG",
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/MetricStreams-
FirehosePutRecords-JN10W9B3",
    "State": "running",
    "CreationDate": "2024-10-11T18:48:59.187000+00:00",
    "LastUpdateDate": "2024-10-11T18:48:59.187000+00:00",
    "OutputFormat": "json",
    "IncludeLinkedAccountsMetrics": false
}
```

Pour plus d'informations, consultez la section Utiliser les flux métriques dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetMetricStreamà la section Référence des AWS CLI commandes.

`get-metric-widget-image`

L'exemple de code suivant montre comment utiliser`get-metric-widget-image`.

AWS CLI

Pour récupérer un graphique instantané de CPUUtilization

L'`get-metric-widget-image`exemple suivant extrait le graphe instantané de la métrique `CPUUtilization` de l' EC2 instance avec l'ID `i-abcde` et enregistre l'image récupérée sous forme de fichier nommé « image.png » sur votre machine locale.

```
aws cloudwatch get-metric-widget-image \
    --metric-widget '{"metrics":[["AWS/EC2","CPUUtilization","InstanceId","i-
abcde"]]}' \
    --output-format png \
    --output text | base64 --decode > image.png
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous GetMetricWidgetImageà la section Référence des AWS CLI commandes.

`list-dashboards`

L'exemple de code suivant montre comment utiliser`list-dashboards`.

AWS CLI

Pour récupérer une liste de tableaux de bord

L'`list-dashboards`exemple suivant répertorie tous les tableaux de bord du compte spécifié.

```
aws cloudwatch list-dashboards
```

Sortie :

```
{
    "DashboardEntries": [
        {
            "DashboardName": "Dashboard-A",
            "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-
A",
            "LastModified": "2024-10-11T18:40:11+00:00",
            "Size": 271
        },
        {
            "DashboardName": "Dashboard-B",
            "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-
B",
```

```
            "LastModified": "2024-10-11T18:44:41+00:00",
            "Size": 522
        }
    ]
}
```

Pour plus d'informations, consultez les CloudWatch tableaux de bord Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListDashboardsà la section Référence des AWS CLI commandes.

`list-metric-streams`

L'exemple de code suivant montre comment utiliser`list-metric-streams`.

AWS CLI

Pour récupérer une liste de flux métriques

L'`list-metric-streams`exemple suivant répertorie tous les flux métriques du compte spécifié.

```
aws cloudwatch list-metric-streams
```

Sortie :

```
{
    "Entries": [
        {
            "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
QuickFull-GuaFbs",
            "CreationDate": "2024-10-11T18:48:59.187000+00:00",
            "LastUpdateDate": "2024-10-11T18:48:59.187000+00:00",
            "Name": "QuickFull-GuaFbs",
            "FirehoseArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG",
            "State": "running",
            "OutputFormat": "json"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utiliser les flux métriques dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListMetricStreamsà la section Référence des AWS CLI commandes.

`list-metrics`

L'exemple de code suivant montre comment utiliser`list-metrics`.

AWS CLI

Pour répertorier les métriques pour Amazon SNS

L'exemple `list-metrics` suivant affiche les métriques pour Amazon SNS.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

Sortie :

```
{
    "Metrics": [
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
```

```
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
```

```
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags associés à une alarme existante*

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées à une alarme nommée demo dans le compte spécifié.

```
aws cloudwatch list-tags-for-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "stack",
```

```
            "Value": "Production"
        },
        {
            "Key": "team",
            "Value": "Devops"
        }
    ]
}
```

Pour plus d'informations, consultez la section Alarmes et balisage dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-anomaly-detector`

L'exemple de code suivant montre comment utiliser`put-anomaly-detector`.

AWS CLI

Pour créer un modèle de détection des anomalies

L'`put-anomaly-detector`exemple suivant crée un modèle de détection des anomalies pour une CloudWatch métrique.

```
aws cloudwatch put-anomaly-detector \
    --namespace AWS/Logs \
    --metric-name IncomingBytes \
    --stat SampleCount
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de la détection des CloudWatch anomalies dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous PutAnomalyDetectorà la section Référence des AWS CLI commandes.

`put-composite-alarm`

L'exemple de code suivant montre comment utiliser`put-composite-alarm`.

AWS CLI

Pour créer une alarme Cloudwatch composite

L'`put-composite-alarm`exemple suivant crée une alarme composite nommée `ProdAlarm` dans le compte spécifié.

```
aws cloudwatch put-composite-alarm \
    --alarm-name ProdAlarm \
    --alarm-rule "ALARM(CPUUtilizationTooHigh) AND ALARM(MemUsageTooHigh)" \
    --alarm-actions arn:aws:sns:us-east-1:123456789012:demo \
    --actions-enabled
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer une alarme composite dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous PutCompositeAlarmà la section Référence des AWS CLI commandes.

`put-dashboard`

L'exemple de code suivant montre comment utiliser`put-dashboard`.

AWS CLI

Pour créer un tableau de bord

L'`put-dashboard`exemple suivant crée un tableau de bord nommé `Dashboard-A` dans le compte spécifié.

```
aws cloudwatch put-dashboard \
    --dashboard-name Dashboard-A \
    --dashboard-body '{"widgets":
[{"height":6,"width":6,"y":0,"x":0,"type":"metric","properties":
{"view":"timeSeries","stacked":false,"metrics":
[["Namespace","CPUUtilization","Environment","Prod","Type","App"]],"region":"us-
east-1"}}]}'
```

Sortie :

```
{
    "DashboardValidationMessages": []
}
```

Pour plus d'informations, consultez la section Création d'un CloudWatch tableau de bord dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous PutDashboardà la section Référence des AWS CLI commandes.

`put-insight-rule`

L'exemple de code suivant montre comment utiliser`put-insight-rule`.

AWS CLI

Pour créer une règle d'analyse des informations sur les contributeurs

L'`put-insight-rule`exemple suivant crée une règle Contributor Insights nommée `VPCFlowLogsContributorInsights` dans le compte spécifié.

```
aws cloudwatch put-insight-rule \
    --rule-name VPCFlowLogsContributorInsights \
    --rule-definition file://insight-rule.json \
    --rule-state ENABLED
```

Contenu de `insight-rule.json` :

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "tcp-flag"
        ]
    },
    "LogFormat": "CLF",
    "LogGroupNames": [
```

```
        "/vpc/flowlogs/*"
    ],
    "Fields": {
        "23": "tcp-flag"
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer une règle Contributor Insights CloudWatch dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous PutInsightRuleà la section Référence des AWS CLI commandes.

`put-metric-alarm`

L'exemple de code suivant montre comment utiliser`put-metric-alarm`.

AWS CLI

Pour envoyer un e-mail Amazon Simple Notification Service lorsque l'utilisation du processeur dépasse 70 %

L'exemple suivant fait appel à la commande `put-metric-alarm` pour envoyer un message e-mail Amazon Simple Notification Service lorsque l'utilisation du processeur dépasse 70 % :

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-
operator GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic --
unit Percent
```

Cette commande revient à l'invite en cas de succès. Si une alarme portant le même nom existe déjà, elle sera remplacée par la nouvelle alarme.

Pour spécifier plusieurs dimensions

L'exemple suivant illustre comment spécifier plusieurs dimensions. Chaque dimension est spécifiée sous forme de Name/Value paire, avec une virgule entre le nom et la valeur. Les différentes dimensions sont séparées par une espace :

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS"
 --metric-name Default_Test --statistic Average --period 60 --evaluation-
periods 3 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --
dimensions Name=key1,Value=value1 Name=key2,Value=value2
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS CLI commandes.

`put-metric-data`

L'exemple de code suivant montre comment utiliser`put-metric-data`.

AWS CLI

Pour publier une métrique personnalisée sur Amazon CloudWatch

L'exemple suivant utilise la `put-metric-data` commande pour publier une métrique personnalisée sur Amazon CloudWatch :

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

Les valeurs de la métrique elle-même sont stockées dans le fichier JSON, `metric.json`.

Voici le contenu de ce fichier :

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Pour plus d'informations, consultez la section Publication de métriques personnalisées dans le manuel Amazon CloudWatch Developer Guide.

Pour spécifier plusieurs dimensions

L'exemple suivant illustre comment spécifier plusieurs dimensions. Chaque dimension est spécifiée sous la forme d'une paire nom=valeur. Les différentes dimensions sont séparées par une virgule :

```
aws cloudwatch put-metric-data --metric-name Buffers --
namespace MyNameSpace --unit Bytes --value 231434333 --
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS CLI commandes.

`put-metric-stream`

L'exemple de code suivant montre comment utiliser`put-metric-stream`.

AWS CLI

Pour créer un flux métrique

L'`put-metric-stream`exemple suivant crée un flux métrique nommé `QuickFull-GuaFb` dans le compte spécifié.

```
aws cloudwatch put-metric-stream \
    --name QuickFull-GuaFbs \
    --firehose-arn arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG \
    --role-arn arn:aws:iam::123456789012:role/service-role/MetricStreams-
FirehosePutRecords-JN10W9B3 \
    --output-format json \
    --no-include-linked-accounts-metrics
```

Sortie :

```
{
    "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/QuickFull-
GuaFbs"
}
```

Pour plus d'informations, consultez la section Configurer un flux métrique dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous PutMetricStreamà la section Référence des AWS CLI commandes.

`set-alarm-state`

L'exemple de code suivant montre comment utiliser`set-alarm-state`.

AWS CLI

Pour modifier temporairement l'état d'une alarme

L'exemple suivant utilise la `set-alarm-state` commande pour modifier temporairement l'état d'une CloudWatch alarme Amazon nommée « myalarm » et la définir sur l'état ALARM à des fins de test :

```
aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-reason "testing purposes"
```

Cette commande revient à l'invite en cas de succès.

- Pour plus de détails sur l'API, reportez-vous SetAlarmStateà la section Référence des AWS CLI commandes.

`start-metric-streams`

L'exemple de code suivant montre comment utiliser`start-metric-streams`.

AWS CLI

Pour démarrer un flux métrique spécifié

L'`start-metric-streams`exemple suivant démarre le flux métrique nommé `QuickFull-GuaFbs` dans le compte spécifié.

```
aws cloudwatch start-metric-streams \
    --names QuickFull-GuaFbs
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les flux métriques dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartMetricStreamsà la section Référence des AWS CLI commandes.

`stop-metric-streams`

L'exemple de code suivant montre comment utiliser`stop-metric-streams`.

AWS CLI

Pour arrêter un flux métrique spécifié

L'`stop-metric-streams`exemple suivant arrête le flux métrique nommé `QuickFull-GuaFbs` dans le compte spécifié.

```
aws cloudwatch stop-metric-streams \
    --names QuickFull-GuaFbs
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les flux métriques dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopMetricStreamsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une ou plusieurs balises à la ressource spécifiée

L'`tag-resource`exemple suivant ajoute deux balises à l'alarme Cloudwatch nommée `demo` dans le compte spécifié.

```
aws cloudwatch tag-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo \
    --tags Key=stack,Value=Production Key=team,Value=Devops
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos CloudWatch ressources Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une ou plusieurs balises de la ressource spécifiée

L'`untag-resource`exemple suivant supprime 2 balises de l'alarme Cloudwatch nommée demo dans le compte spécifié.

```
aws cloudwatch untag-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo \
    --tag-keys stack team
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos CloudWatch ressources Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

CloudWatch Exemples de journaux utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with CloudWatch Logs.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-log-group`

L'exemple de code suivant montre comment utiliser`create-log-group`.

AWS CLI

La commande suivante crée un groupe de journaux nommé `my-logs` :

```
aws logs create-log-group --log-group-name my-logs
```

- Pour plus de détails sur l'API, reportez-vous CreateLogGroupà la section Référence des AWS CLI commandes.

`create-log-stream`

L'exemple de code suivant montre comment utiliser`create-log-stream`.

AWS CLI

La commande suivante crée un flux de journal nommé `20150601` dans le groupe de journaux `my-logs` :

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Pour plus de détails sur l'API, reportez-vous CreateLogStreamà la section Référence des AWS CLI commandes.

`delete-log-group`

L'exemple de code suivant montre comment utiliser`delete-log-group`.

AWS CLI

La commande suivante supprime un groupe de journaux nommé `my-logs` :

```
aws logs delete-log-group --log-group-name my-logs
```

- Pour plus de détails sur l'API, reportez-vous DeleteLogGroupà la section Référence des AWS CLI commandes.

`delete-log-stream`

L'exemple de code suivant montre comment utiliser`delete-log-stream`.

AWS CLI

La commande suivante supprime un flux de journaux nommé `20150531` à partir d'un groupe de journaux nommé `my-logs` :

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- Pour plus de détails sur l'API, reportez-vous DeleteLogStreamà la section Référence des AWS CLI commandes.

`delete-retention-policy`

L'exemple de code suivant montre comment utiliser`delete-retention-policy`.

AWS CLI

La commande suivante supprime la politique de rétention précédemment appliquée à un groupe de journaux nommé `my-logs` :

```
aws logs delete-retention-policy --log-group-name my-logs
```

- Pour plus de détails sur l'API, reportez-vous DeleteRetentionPolicyà la section Référence des AWS CLI commandes.

`describe-log-groups`

L'exemple de code suivant montre comment utiliser`describe-log-groups`.

AWS CLI

La commande suivante décrit un groupe de journaux nommé `my-logs` :

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Sortie :

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLogGroupsà la section Référence des AWS CLI commandes.

`describe-log-streams`

L'exemple de code suivant montre comment utiliser`describe-log-streams`.

AWS CLI

La commande suivante affiche tous les flux de journaux `2015` en commençant par le préfixe du groupe `my-logs` de journaux :

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

Sortie :

```
{
    "logStreams": [
        {
            "creationTime": 1433189871774,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150531",
            "logStreamName": "20150531",
```

```
            "storedBytes": 0
        },
        {
            "creationTime": 1433189873898,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150601",
            "logStreamName": "20150601",
            "storedBytes": 0
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLogStreamsà la section Référence des AWS CLI commandes.

`get-log-events`

L'exemple de code suivant montre comment utiliser`get-log-events`.

AWS CLI

La commande suivante extrait les événements du journal à partir d'un flux de journaux nommé `20150601` dans le groupe `my-logs` de journaux :

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Sortie :

```
{
    "nextForwardToken":
 "f/31961209122447488583055879464742346735121166569214640130",
    "events": [
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190516679,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
```

```
        },
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184358,
            "message": "Example Event 2"
        }
    ],
    "nextBackwardToken":
 "b/31961209122358285602261756944988674324553373268216709120"
}
```

- Pour plus de détails sur l'API, reportez-vous GetLogEventsà la section Référence des AWS CLI commandes.

`put-log-events`

L'exemple de code suivant montre comment utiliser`put-log-events`.

AWS CLI

La commande suivante place les événements du journal dans un flux de journaux nommé `20150601` dans le groupe de journaux `my-logs` :

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-
events file://events
```

Sortie :

```
{
    "nextSequenceToken": "495426724868310740095796045676567882148068632824696073 46"
}
```

L'exemple ci-dessus lit un tableau JSON d'événements à partir d'un fichier nommé `events` dans le répertoire courant :

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
```

```
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

Chaque appel suivant nécessite que le jeton de séquence suivant fourni par l'appel précédent soit spécifié avec l'option de jeton de séquence :

```
aws logs put-log-events --log-group-name my-logs --log-
stream-name 20150601 --log-events file://events2 --sequence-
token "49542672486831074009579604567656788214806863282469607346"
```

Sortie :

```
{
    "nextSequenceToken": "49542672486831074009579604567900991230369019956308219826"
}
```

- Pour plus de détails sur l'API, reportez-vous PutLogEventsà la section Référence des AWS CLI commandes.

`put-retention-policy`

L'exemple de code suivant montre comment utiliser`put-retention-policy`.

AWS CLI

La commande suivante ajoute une politique de conservation de 5 jours à un groupe de journaux nommé `my-logs` :

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- Pour plus de détails sur l'API, reportez-vous PutRetentionPolicyà la section Référence des AWS CLI commandes.

CloudWatch Exemples de surveillance réseau à l'aide de AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de la surveillance du CloudWatch réseau.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-monitor`

L'exemple de code suivant montre comment utiliser`create-monitor`.

AWS CLI

Exemple 1 : pour créer un moniteur réseau avec une période d'agrégation

L'`create-monitor`exemple suivant crée un moniteur dont le nom `Example_NetworkMonitor` est `aggregationPeriod` défini sur `30` secondes. L'initiale `state` du moniteur sera due au `INACTIVE` fait qu'aucune sonde n'y est associée. L'état passe `ACTIVE` uniquement lorsque des sondes sont ajoutées. Vous pouvez utiliser les commandes update-monitor ou create-probe pour ajouter des sondes à ce moniteur.

```
aws networkmonitor create-monitor \
     --monitor-name Example_NetworkMonitor \
     --aggregation-period 30
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
```

```
    "monitorName": "Example_NetworkMonitor",
    "state": "INACTIVE",
    "aggregationPeriod": 30,
    "tags": {}
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

Exemple 2 : pour créer un moniteur réseau avec une sonde utilisant le protocole TCP et incluant également des balises

L'`create-monitor`exemple suivant crée un moniteur nomméExample_NetworkMonitor. La commande crée également une sonde qui utilise le ICMP protocole et inclut des balises. Puisque aucun n'aggregationPeriodest transmis dans la demande, les 60 secondes sont définies par défaut. Le moniteur avec `state` la sonde durera PENDING jusqu'à ce que le moniteur soit en placeACTIVE. Cela peut prendre plusieurs minutes. À ce moment-là, ils `state` deviendrontACTIVE, et vous pourrez commencer à consulter CloudWatch les statistiques.

```
aws networkmonitor create-monitor \
    --monitor-name Example_NetworkMonitor \
    --probes sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,probeTags={Name=Prob
 \
    --tags Monitor=Monitor1
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 60,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

Exemple 3 : pour créer un moniteur réseau avec une sonde utilisant le protocole ICMP et incluant également des balises

L'`create-monitor`exemple suivant crée un moniteur nommé `Example_NetworkMonitor` avec un nombre `aggregationPeriod` de `30` secondes. La commande crée également une sonde qui utilise le `ICMP` protocole et inclut des balises. Puisque aucun n'`aggregationPeriod`est transmis dans la demande, les `60` secondes sont définies par défaut. Le moniteur avec `state` la sonde durera `PENDING` jusqu'à ce que le moniteur soit en place`ACTIVE`. Cela peut prendre plusieurs minutes. À ce moment-là, ils `state` deviendront`ACTIVE`, et vous pourrez commencer à consulter CloudWatch les statistiques.

```
aws networkmonitor create-monitor \
     --monitor-name Example_NetworkMonitor \
     --aggregation-period 30 \
     --probes sourceArn=arn:aws:ec2:region111122223333:subnet/subnet-
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,probeTags={Name=Probe1} \
     --tags Monitor=Monitor1
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 30,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateMonitorà la section Référence des AWS CLI commandes.

`create-probe`

L'exemple de code suivant montre comment utiliser`create-probe`.

AWS CLI

Exemple 1 : pour créer une sonde utilisant le protocole TCP et l'ajouter à un moniteur réseau

L'`create-probe`exemple suivant crée une sonde qui utilise `TCP` `protocol` et ajoute la sonde à un moniteur nomméExample_NetworkMonitor. Une fois créé, `state` le moniteur avec la sonde durera `PENDING` jusqu'à ce que le moniteur soitACTIVE. Cela peut prendre plusieurs minutes, puis l'état passera à l'étatACTIVE, et vous pourrez commencer à consulter CloudWatch les statistiques.

```
aws networkmonitor create-probe \
    --monitor-name Example_NetworkMonitor \
    --probe sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,tags={Name=Probe1}
```

Sortie :

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",
    "destination": "10.0.0.100",
    "destinationPort": 80,
    "packetSize": 56,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "PENDING",
    "createdAt": "2024-03-29T12:41:57.314000-04:00",
    "modifiedAt": "2024-03-29T12:41:57.314000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

Exemple 2 : pour créer une sonde utilisant une sonde à l'aide de l'ICMP et pour l'ajouter à un moniteur réseau

L'`create-probe`exemple suivant crée une sonde qui utilise `ICMP` `protocol` et ajoute la sonde à un moniteur nomméExample_NetworkMonitor. Une fois créé, `state` le moniteur avec la sonde durera `PENDING` jusqu'à ce que le moniteur soitACTIVE. Cela peut prendre plusieurs minutes, puis l'état passera à l'étatACTIVE, et vous pourrez commencer à consulter CloudWatch les statistiques.

```
aws networkmonitor create-probe \
    --monitor-name Example_NetworkMonitor \
    --probe sourceArn=arn:aws:ec2:region:012345678910:subnet/subnet-
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,tags={Name=Probe1}
```

Sortie :

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",
    "destination": "10.0.0.100",
    "packetSize": 56,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "PENDING",
    "createdAt": "2024-03-29T12:44:02.452000-04:00",
    "modifiedAt": "2024-03-29T12:44:02.452000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateProbeà la section Référence des AWS CLI commandes.

`delete-monitor`

L'exemple de code suivant montre comment utiliser`delete-monitor`.

AWS CLI

Pour supprimer un moniteur

L'`delete-monitor`exemple suivant supprime un moniteur nomméExample_NetworkMonitor.

```
aws networkmonitor delete-monitor \
    --monitor-name Example_NetworkMonitor
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteMonitorà la section Référence des AWS CLI commandes.

`delete-probe`

L'exemple de code suivant montre comment utiliser`delete-probe`.

AWS CLI

Pour supprimer une sonde

L'`delete-probe`exemple suivant supprime une sonde portant l'ID `probe-12345` d'un moniteur réseau nomméExample_NetworkMonitor.

```
aws networkmonitor delete-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteProbeà la section Référence des AWS CLI commandes.

`get-monitor`

L'exemple de code suivant montre comment utiliser`get-monitor`.

AWS CLI

Pour obtenir des informations sur le moniteur

L'`get-monitor`exemple suivant permet d'obtenir des informations sur un moniteur nomméExample_NetworkMonitor.

```
aws networkmonitor get-monitor \
```

```
    --monitor-name Example_NetworkMonitor
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "ACTIVE",
    "aggregationPeriod": 60,
    "tags": {},
    "probes": [],
    "createdAt": "2024-04-01T17:58:07.211000-04:00",
    "modifiedAt": "2024-04-01T17:58:07.211000-04:00"
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetMonitorà la section Référence des AWS CLI commandes.

`get-probe`

L'exemple de code suivant montre comment utiliser`get-probe`.

AWS CLI

Pour afficher les détails de la sonde

L'`get-probe`exemple suivant renvoie les détails d'une sonde associée à un moniteur nomméExample_NetworkMonitor. probeID probe-12345

```
aws networkmonitor get-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345
```

Sortie :

```
{
    "probeId": "probe-12345",
```

```
    "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",
    "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",
    "destination": "10.0.0.100",
    "destinationPort": 80,
    "protocol": "TCP",
    "packetSize": 56,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "ACTIVE",
    "createdAt": "2024-03-29T12:41:57.314000-04:00",
    "modifiedAt": "2024-03-29T12:42:28.610000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetProbeà la section Référence des AWS CLI commandes.

`list-monitors`

L'exemple de code suivant montre comment utiliser`list-monitors`.

AWS CLI

Exemple 1 : pour répertorier tous les moniteurs (moniteur unique)

L'`list-monitors`exemple suivant renvoie une liste d'un seul moniteur. Le moniteur l'`state`est `ACTIVE` et il a une durée `aggregationPeriod` de 60 secondes.

```
aws networkmonitor list-monitors
```

Sortie :

```
{
    "monitors": [{
            "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
            "monitorName": "Example_NetworkMonitor",
```

```
            "state": "ACTIVE",
            "aggregationPeriod": 60,
            "tags": {
                "Monitor": "Monitor1"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

Exemple 2 : pour répertorier tous les moniteurs (plusieurs moniteurs)

L'`list-monitors`exemple suivant renvoie une liste de trois moniteurs. Le `state` d'un moniteur est `ACTIVE` et génère des CloudWatch métriques. Les états des deux autres moniteurs ne `INACTIVE` génèrent pas de CloudWatch métriques. Les trois moniteurs utilisent une durée `aggregationPeriod` de 60 secondes.

```
aws networkmonitor list-monitors
```

Sortie :

```
{
    "monitors": [
        {
            "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor",
            "monitorName": "Example_NetworkMonitor",
            "state": "INACTIVE",
            "aggregationPeriod": 60,
            "tags": {}
        },
        {
            "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor2",
            "monitorName": "Example_NetworkMonitor2",
            "state": "ACTIVE",
            "aggregationPeriod": 60,
            "tags": {
                "Monitor": "Monitor1"
            }
```

```
        },
        {
            "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
TestNetworkMonitor_CLI",
            "monitorName": "TestNetworkMonitor_CLI",
            "state": "INACTIVE",
            "aggregationPeriod": 60,
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListMonitorsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant renvoie la liste des balises d'un moniteur nomméExample_NetworkMonitor.

```
aws networkmonitor list-tags-for-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor
```

Sortie :

```
{
    "tags": {
        "Environment": "Dev",
        "Application": "PetStore"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant balise un moniteur nommé `Example_NetworkMonitor` avec `Environment=Dev` et `Application=PetStore` tags.

```
aws networkmonitor tag-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor \
    --tags Environment=Dev,Application=PetStore
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour annuler le balisage d'une ressource

L'`untag-resource`exemple suivant supprime un `tag-keys` paramètre avec la paire clé-valeur de `Environment Application` de son association avec un moniteur nommé. `Example_NetworkMonitor`

```
aws networkmonitor untag-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor \
    --tag-keys Environment Application
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-monitor`

L'exemple de code suivant montre comment utiliser`update-monitor`.

AWS CLI

Pour mettre à jour un moniteur

L'`update-monitor`exemple suivant fait passer la valeur d'un moniteur `aggregationPeriod` de `60` quelques secondes à `30` quelques secondes.

```
aws networkmonitor update-monitor \
    --monitor-name Example_NetworkMonitor \
    --aggregation-period 30
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 30,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateMonitorà la section Référence des AWS CLI commandes.

`update-probe`

L'exemple de code suivant montre comment utiliser`update-probe`.

AWS CLI

Pour mettre à jour une sonde

L'`update-probe`exemple suivant met à jour l'adresse `destination` IP d'origine d'une sonde et met également à jour le `packetSize` vers`60`.

```
aws networkmonitor update-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345 \
    --destination 10.0.0.150 \
    --packet-size 60
```

Sortie :

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",
    "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",
    "destination": "10.0.0.150",
    "destinationPort": 80,
    "protocol": "TCP",
    "packetSize": 60,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "PENDING",
    "createdAt": "2024-03-29T12:41:57.314000-04:00",
    "modifiedAt": "2024-03-29T13:52:23.115000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Network Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateProbeà la section Référence des AWS CLI commandes.

CloudWatch Exemples d'Observability Access Monitor utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du moniteur AWS Command Line Interface d'accès à l' CloudWatch observabilité.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-link`

L'exemple de code suivant montre comment utiliser`create-link`.

AWS CLI

Pour créer un lien

L'`create-link`exemple suivant crée un lien entre un compte source et un récepteur que vous avez créé dans un compte de surveillance.

```
aws oam create-link \
    --label-template sourceAccount \
    --resource-types AWS::CloudWatch::Metric \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

Sortie :

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-
example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "Tags": {}
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateLinkà la section Référence des AWS CLI commandes.

`create-sink`

L'exemple de code suivant montre comment utiliser`create-sink`.

AWS CLI

Pour créer un évier

L'`create-sink`exemple suivant crée un puits dans le compte courant, afin qu'il puisse être utilisé comme compte de surveillance dans le cadre de l' CloudWatch observabilité entre comptes.

```
aws oam create-sink \
    --name DemoSink
```

Sortie :

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
```

```
    "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Name": "DemoSink",
    "Tags": {}
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateSinkà la section Référence des AWS CLI commandes.

`delete-link`

L'exemple de code suivant montre comment utiliser`delete-link`.

AWS CLI

Pour supprimer un lien

L'`delete-link`exemple suivant supprime un lien entre un récepteur de comptes de surveillance et un compte source.

```
aws oam delete-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteLinkà la section Référence des AWS CLI commandes.

`delete-sink`

L'exemple de code suivant montre comment utiliser`delete-sink`.

AWS CLI

Pour supprimer un évier

L'`delete-sink`exemple suivant supprime un évier. Vous devez supprimer tous les liens vers un récepteur avant de pouvoir supprimer ce dernier.

```
aws oam delete-sink \
    --identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteSinkà la section Référence des AWS CLI commandes.

`get-link`

L'exemple de code suivant montre comment utiliser`get-link`.

AWS CLI

Pour renvoyer des informations complètes sur un lien

L'`get-link`exemple suivant renvoie des informations complètes sur un lien.

```
aws oam get-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111
```

Sortie :

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
```

```
    "Tags": {}
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetLinkà la section Référence des AWS CLI commandes.

`get-sink-policy`

L'exemple de code suivant montre comment utiliser`get-sink-policy`.

AWS CLI

Pour renvoyer la politique de récepteur actuelle attachée au récepteur

L'`get-sink-policy`exemple suivant renvoie la politique de récepteur actuelle attachée au récepteur.

```
aws oam get-sink-policy \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

Sortie :

```
{
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "SinkId": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789111:root\"},\"Action\":[\"oam:CreateLink\",\"oam:UpdateLink\"],\"Resource\":\"*\",\"Condition\":{\"ForAllValues:StringEquals\":{\"oam:ResourceTypes\":[\"AWS::Logs::LogGroup\",\"AWS::CloudWatch::Metric\",\"AWS::XRay::Trace\",\"AWS::ApplicationInsights::Application\"]}}}]}"
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetSinkPolicyà la section Référence des AWS CLI commandes.

`get-sink`

L'exemple de code suivant montre comment utiliser`get-sink`.

AWS CLI

Pour renvoyer des informations complètes sur un compte de surveillance, réceptacle

L'`get-sink`exemple suivant renvoie des informations complètes sur un récepteur de compte de surveillance.

```
aws oam get-sink \
    --identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

Sortie :

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Name": "DemoSink",
    "Tags": {}
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetSinkà la section Référence des AWS CLI commandes.

`list-attached-links`

L'exemple de code suivant montre comment utiliser`list-attached-links`.

AWS CLI

Pour renvoyer une liste des liens de comptes sources liés à ce compte de surveillance, réceptacle

L'`list-attached-links`exemple suivant renvoie une liste de liens de comptes source liés à ce récepteur de comptes de surveillance.

```
aws oam list-attached-links \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

Sortie :

```
{
    "Items": [{
        "Label": "Monitoring account",
        "LinkArn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
        "ResourceTypes": [
            "AWS::ApplicationInsights::Application",
            "AWS::Logs::LogGroup",
            "AWS::CloudWatch::Metric",
            "AWS::XRay::Trace"
        ]
    }]
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListAttachedLinksà la section Référence des AWS CLI commandes.

`list-links`

L'exemple de code suivant montre comment utiliser`list-links`.

AWS CLI

Pour renvoyer une liste de liens pour un compte de surveillance, réceptacle

L'`list-links`exemple suivant renvoie une liste de liens pour un récepteur de comptes de surveillance. Exécutez cette opération dans un compte source pour renvoyer une liste de liens vers les récepteurs de compte de surveillance que possède ce compte source.

```
aws oam list-links
```

Sortie :

```
{
    "Items": [{
        "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-
example11111",
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Label": "sourceAccount",
        "ResourceTypes": [
            "AWS::CloudWatch::Metric"
        ],
        "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345"
    }]
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListLinksà la section Référence des AWS CLI commandes.

`list-sinks`

L'exemple de code suivant montre comment utiliser`list-sinks`.

AWS CLI

Pour renvoyer la liste des puits créés dans le compte de surveillance

L'`list-sinks`exemple suivant renvoie une liste de récepteurs créés dans le compte de surveillance. Exécutez cette opération dans un compte de surveillance.

```
aws oam list-sinks
```

Sortie :

```
{
    "Items": [
```

```
        {
            "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
            "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
            "Name": "DemoSink"
        }
    ]
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListSinksà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour afficher les balises associées à une ressource

L'`list-tags-for-resource`exemple suivant affiche les balises associées à un récepteur.

```
aws oam list-tags-for-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345
```

Sortie :

```
{
    "Tags": {
        "Team": "Devops"
    }
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-sink-policy`

L'exemple de code suivant montre comment utiliser`put-sink-policy`.

AWS CLI

Pour créer ou mettre à jour la politique de ressources

L'`put-sink-policy`exemple suivant crée la politique de ressources qui autorise les comptes source à établir un lien avec le récepteur de comptes de surveillance.

```
aws oam put-sink-policy \
    --policy '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::123456789111:root"},"Action":
["oam:CreateLink","oam:UpdateLink"],"Resource":"*","Condition":
{"ForAllValues:StringEquals":{"oam:ResourceTypes":
["AWS::Logs::LogGroup","AWS::CloudWatch::Metric","AWS::XRay::Trace","AWS::ApplicationInsight
 \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-
cdef-example12345
```

Sortie :

```
{
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "SinkId": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789111:root\"},
\"Action\":[\"oam:CreateLink\",\"oam:UpdateLink\"],\"Resource\":\"*\",
\"Condition\":{\"ForAllValues:StringEquals\":{\"oam:ResourceTypes\":
[\"AWS::Logs::LogGroup\",\"AWS::CloudWatch::Metric\",\"AWS::XRay::Trace\",
\"AWS::ApplicationInsights::Application\"]}}}]}"
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous PutSinkPolicyà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour attribuer une ou plusieurs balises à la ressource spécifiée

L'`tag-resource`exemple suivant balise un évier`arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345`.

```
aws oam tag-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345 \
    --tags team=Devops
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une ou plusieurs balises de la ressource spécifiée.

L'`untag-resource`exemple suivant supprime une balise contenant la clé `team` du récepteur`arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345`.

```
aws oam untag-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/f3f42f60-f0f2-425c-1234-12347bdd821f \
    --tag-keys team
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-link`

L'exemple de code suivant montre comment utiliser`update-link`.

AWS CLI

Pour modifier les types de données partagés entre un compte source et son compte de surveillance associé (récepteur)

L'`update-link`exemple suivant met à jour le lien `arn:aws:oam:us-east-2:123456789111:link/0123e691-e7ef-43fa-1234-c57c837fced0` avec les types de ressources `AWS::CloudWatch::Metric` et`AWS::Logs::LogGroup`.

```
aws oam update-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111 \
    --resource-types "AWS::CloudWatch::Metric" "AWS::Logs::LogGroup"
```

Sortie :

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric",
        "AWS::Logs::LogGroup"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "Tags": {}
}
```

Pour plus d'informations, consultez la CloudWatch section sur l'observabilité entre comptes dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateLinkà la section Référence des AWS CLI commandes.

CloudWatch Exemples d'administration de l'observabilité utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l'administrateur AWS Command Line Interface d' CloudWatch observabilité.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-telemetry-evaluation-status-for-organization`

L'exemple de code suivant montre comment utiliser`get-telemetry-evaluation-status-for-organization`.

AWS CLI

Pour obtenir le statut d'intégration de l'organisation à la télémétrie

L'`get-telemetry-evaluation-status-for-organization`exemple suivant renvoie l'état d'intégration actuel de la fonctionnalité de configuration de télémétrie pour l'organisation.

```
aws observabilityadmin get-telemetry-evaluation-status-for-organization
```

Sortie :

```
{
    "Status": "RUNNING"
}
```

Pour plus d'informations, consultez la section Audit des configurations de CloudWatch télémétrie dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetTelemetryEvaluationStatusForOrganizationà la section Référence des AWS CLI commandes.

`get-telemetry-evaluation-status`

L'exemple de code suivant montre comment utiliser`get-telemetry-evaluation-status`.

AWS CLI

Pour connaître le statut d'intégration de la télémétrie au compte

L'`get-telemetry-evaluation-status`exemple suivant renvoie l'état d'intégration actuel de la fonctionnalité de configuration de télémétrie dans le compte spécifié.

```
aws observabilityadmin get-telemetry-evaluation-status
```

Sortie :

```
{
    "Status": "RUNNING"
}
```

Pour plus d'informations, consultez la section Audit des configurations de CloudWatch télémétrie dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetTelemetryEvaluationStatusà la section Référence des AWS CLI commandes.

`list-resource-telemetry-for-organization`

L'exemple de code suivant montre comment utiliser`list-resource-telemetry-for-organization`.

AWS CLI

Pour récupérer les configurations de télémétrie de l'organisation

L'`list-resource-telemetry-for-organization`exemple suivant renvoie une liste des configurations de télémétrie de l'organisation pour les AWS ressources prises en charge par la configuration de télémétrie.

```
aws observabilityadmin list-resource-telemetry-for-organization \
    --resource-types  AWS::EC2::Instance
```

Sortie :

```
{
    "TelemetryConfigurations": [
        {
            "AccountIdentifier": "111111111111",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-a166400b",
            "ResourceTags": {
                "Name": "dev"
            },
            "LastUpdateTimeStamp": 1733168548521
        },
        {
            "AccountIdentifier": "222222222222",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-b188560f",
            "ResourceTags": {
                "Name": "apache"
            },
            "LastUpdateTimeStamp": 1732744260182
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Audit des configurations de CloudWatch télémétrie dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListResourceTelemetryForOrganizationà la section Référence des AWS CLI commandes.

`list-resource-telemetry`

L'exemple de code suivant montre comment utiliser`list-resource-telemetry`.

AWS CLI

Pour récupérer les configurations de télémétrie du compte

L'`list-resource-telemetry`exemple suivant renvoie une liste de configurations de télémétrie pour les AWS ressources prises en charge par la configuration de télémétrie dans le compte spécifié.

```
aws observabilityadmin list-resource-telemetry \
    --resource-types  AWS::EC2::Instance
```

Sortie :

```
{
    "TelemetryConfigurations": [
        {
            "AccountIdentifier": "111111111111",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-0e979d278b040f856",
            "ResourceTags": {
                "Name": "apache"
            },
            "LastUpdateTimeStamp": 1732744260182
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Audit des configurations de CloudWatch télémétrie dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListResourceTelemetryà la section Référence des AWS CLI commandes.

`start-telemetry-evaluation-for-organization`

L'exemple de code suivant montre comment utiliser`start-telemetry-evaluation-for-organization`.

AWS CLI

Pour activer la fonctionnalité de configuration de télémétrie

L'`start-telemetry-evaluation-for-organization`exemple suivant active la fonctionnalité de configuration de télémétrie pour l'organisation.

```
aws observabilityadmin start-telemetry-evaluation-for-organization
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de l'audit CloudWatch télémétrique dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartTelemetryEvaluationForOrganizationà la section Référence des AWS CLI commandes.

`start-telemetry-evaluation`

L'exemple de code suivant montre comment utiliser`start-telemetry-evaluation`.

AWS CLI

Pour activer la fonctionnalité de configuration de télémétrie

L'`start-telemetry-evaluation`exemple suivant active la fonctionnalité de configuration de télémétrie dans le compte spécifié.

```
aws observabilityadmin start-telemetry-evaluation
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de l'audit CloudWatch télémétrique dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartTelemetryEvaluationà la section Référence des AWS CLI commandes.

`stop-telemetry-evaluation-for-organization`

L'exemple de code suivant montre comment utiliser`stop-telemetry-evaluation-for-organization`.

AWS CLI

Pour désactiver la fonctionnalité de configuration de télémétrie

L'`stop-telemetry-evaluation-for-organization`exemple suivant désactive la fonctionnalité de configuration de télémétrie pour l'organisation.

```
aws observabilityadmin stop-telemetry-evaluation-for-organization
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation de l'audit CloudWatch télémétrique dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopTelemetryEvaluationForOrganizationà la section Référence des AWS CLI commandes.

`stop-telemetry-evaluation`

L'exemple de code suivant montre comment utiliser`stop-telemetry-evaluation`.

AWS CLI

Pour désactiver la fonctionnalité de configuration de télémétrie

L'`stop-telemetry-evaluation`exemple suivant désactive la fonctionnalité de configuration de télémétrie dans le compte spécifié.

```
aws observabilityadmin stop-telemetry-evaluation
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation de l'audit CloudWatch télémétrique dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopTelemetryEvaluationà la section Référence des AWS CLI commandes.

CloudWatch Exemples de produits Synthetics utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with CloudWatch Synthetics.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-resource`

L'exemple de code suivant montre comment utiliser`associate-resource`.

AWS CLI

Pour associer un canari à un groupe

L'`associate-resource`exemple suivant associe un canari à un groupe nommé`demo_group`.

```
aws synthetics associate-resource \
    --group-identifier demo_group \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir AssociateResourcela section Référence des AWS CLI commandes.

`create-canary`

L'exemple de code suivant montre comment utiliser`create-canary`.

AWS CLI

Pour créer un canari

L'`create-canary`exemple suivant crée un canari nommé`demo_canary`.

```
aws synthetics create-canary \
    --name demo_canary \
    --code '{"S3Bucket": "artifacts3bucket", "S3Key":"demo_canary.zip", "Handler":
 "index.lambda_handler"}' \
    --artifact-s3-location s3://amzn-s3-demo-bucket/demo_canary.zip \
    --execution-role-arn arn:aws:iam::123456789012:role/demo_canary_role \
    --schedule Expression="rate(10 minutes)" \
    --runtime-version syn-nodejs-puppeteer-9.1
```

Sortie :

```
{
    "Canary": {
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Name": "demo_canary",
        "Code": {
            "Handler": "index.lambda_handler"
        },
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/demo_canary_role",
        "Schedule": {
            "Expression": "rate(10 minutes)",
            "DurationInSeconds": 0
        },
        "RunConfig": {
            "TimeoutInSeconds": 600,
```

```
            "MemoryInMB": 1000,
            "ActiveTracing": false
        },
        "SuccessRetentionPeriodInDays": 31,
        "FailureRetentionPeriodInDays": 31,
        "Status": {
            "State": "CREATING",
            "StateReasonCode": "CREATE_PENDING"
        },
        "Timeline": {
            "Created": "2024-10-15T19:03:08.826000+05:30",
            "LastModified": "2024-10-15T19:03:08.826000+05:30"
        },
        "ArtifactS3Location": "amzn-s3-demo-bucket/demo_canary.zip",
        "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
        "Tags": {}
    }
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir CreateCanaryla section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour créer un groupe

L'`create-group`exemple suivant crée un groupe nommé`demo_group`.

```
aws synthetics create-group \
    --name demo_group
```

Sortie :

```
{
    "Group": {
        "Id": "example123",
```

```
        "Name": "demo_group",
        "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123",
        "Tags": {},
        "CreatedTime": "2024-10-15T14:47:23.811000+05:30",
        "LastModifiedTime": "2024-10-15T14:47:23.811000+05:30"
    }
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir CreateGroupla section Référence des AWS CLI commandes.

`delete-canary`

L'exemple de code suivant montre comment utiliser`delete-canary`.

AWS CLI

Pour supprimer définitivement un canari

L'`delete-canary`exemple suivant supprime un canari nommé`demo_canary`.

```
aws synthetics delete-canary \
    --name demo_canary
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir DeleteCanaryla section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe

L'`delete-group`exemple suivant supprime un groupe nommé`demo_group`.

```
aws synthetics delete-group \
    --group-identifier demo_group
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir DeleteGroupla section Référence des AWS CLI commandes.

`describe-canaries-last-run`

L'exemple de code suivant montre comment utiliser`describe-canaries-last-run`.

AWS CLI

Pour consulter les informations relatives à la dernière édition de chaque canari

L'`describe-canaries-last-run`exemple suivant renvoie la dernière exécution de chaque canari que vous avez créé.

```
aws synthetics describe-canaries-last-run
```

Sortie :

```
{
    "CanariesLastRun": [
        {
            "CanaryName": "demo_canary",
            "LastRun": {
                "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
                "Name": "demo_canary",
                "Status": {
                    "State": "PASSED",
                    "StateReason": "",
                    "StateReasonCode": ""
                },
                "Timeline": {
                    "Started": "2024-10-15T19:20:39.691000+05:30",
                    "Completed": "2024-10-15T19:20:58.211000+05:30"
                },
```

```
                "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/
us-east-1/demo_canary-abc-example1234/2024/10/15/13/50-39-690"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir DescribeCanariesLastRunla section Référence des AWS CLI commandes.

`describe-canaries`

L'exemple de code suivant montre comment utiliser`describe-canaries`.

AWS CLI

Pour répertorier les canaris dans votre compte

L'`describe-canaries`exemple suivant répertorie les informations relatives aux canaris de votre compte.

```
aws synthetics describe-canaries
```

Sortie :

```
{
    "Canaries": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
            "Name": "demo_canary",
            "Code": {
                "SourceLocationArn": "arn:aws:lambda:us-
east-1:123456789012:layer:cwsyn-demo_canary-a1b2c3d4-5678-90ab-cdef-
example11111b8:1",
                "Handler": "pageLoadBlueprint.handler"
            },
            "ExecutionRoleArn": "arn:aws:iam::123456789012:role/service-role/
CloudWatchSyntheticsRole-demo_canary-a12-a123bc456789",
            "Schedule": {
```

```
                "Expression": "rate(5 minutes)",
                "DurationInSeconds": 0
            },
            "RunConfig": {
                "TimeoutInSeconds": 300,
                "MemoryInMB": 1000,
                "ActiveTracing": false
            },
            "SuccessRetentionPeriodInDays": 31,
            "FailureRetentionPeriodInDays": 31,
            "Status": {
            "State": "RUNNING"
            },
            "Timeline": {
                "Created": "2024-10-15T18:55:15.168000+05:30",
                "LastModified": "2024-10-15T18:55:40.540000+05:30",
                "LastStarted": "2024-10-15T18:55:40.540000+05:30"
            },
            "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-a12-a123bc456789",
            "EngineArn": "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
            "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
            "Tags": {
                "blueprint": "heartbeat"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir DescribeCanariesla section Référence des AWS CLI commandes.

`describe-runtime-versions`

L'exemple de code suivant montre comment utiliser`describe-runtime-versions`.

AWS CLI

Pour renvoyer une liste des versions synthétiques de Canary Runtime

L'`describe-runtime-versions`exemple suivant renvoie la liste des versions d'exécution de Synthetics Canary.

```
aws synthetics describe-runtime-versions
```

Sortie :

```
{
    "RuntimeVersions": [
        {
            "VersionName": "syn-nodejs-puppeteer-9.1",
            "Description": "Security fixes and bug fix for date range error in har.
 Dependencies: Node JS 20.x, Puppeteer-core 22.12.1, Chromium 126.0.6478.126",
            "ReleaseDate": "2024-10-02T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-9.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 20.x, Puppeteer-core 22.12.1, Chromium 126.0.6478.126",
            "ReleaseDate": "2024-07-22T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-8.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 20.x, Puppeteer-core 22.10.0, Chromium 125.0.6422.112",
            "ReleaseDate": "2024-06-21T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-7.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 18.x, Puppeteer-core 21.9.0, Chromium 121.0.6167.139",
            "ReleaseDate": "2024-03-08T05:30:00+05:30"
            },
        {
            "VersionName": "syn-nodejs-puppeteer-6.2",
            "Description": "Updated shared libraries for Chromium and added
 ephemeral storage monitoring. Dependencies: Node JS 18.x, Puppeteer-core 19.7.0,
 Chromium 111.0.5563.146",
            "ReleaseDate": "2024-02-02T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-6.1",
```

```
            "Description": "Added puppeteer launch retry. Dependencies: Node JS
 18.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-11-13T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-6.0",
            "Description": "Reduced X-Ray traces of a canary run, improved duration
 metric and upgraded to NodeJS 18.x. Dependencies: Node JS 18.x, Puppeteer-core
 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-09-15T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.2",
            "Description": "Updated shared libraries for Chromium. Dependencies:
 Node JS 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2024-02-01T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.1",
            "Description": "Fixes a bug about missing request headers in har.
 Dependencies: Node JS 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-08-09T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.0",
            "Description": "Upgraded Puppeteer and Chromium. Dependencies: Node JS
 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-07-21T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-4.0",
            "Description": "Upgraded to NodeJS 16.x. Dependencies: Node JS 16.x,
 Puppeteer-core 5.5.0, Chromium 92.0.4512.0",
            "ReleaseDate": "2023-05-01T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir DescribeRuntimeVersionsla section Référence des AWS CLI commandes.

`disassociate-resource`

L'exemple de code suivant montre comment utiliser`disassociate-resource`.

AWS CLI

Pour supprimer un canari d'un groupe

L'`disassociate-resource`exemple suivant supprime un canari du groupe nommé`demo_group`.

```
aws synthetics disassociate-resource \
    --group-identifier demo_group \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir DisassociateResourcela section Référence des AWS CLI commandes.

`get-canary-runs`

L'exemple de code suivant montre comment utiliser`get-canary-runs`.

AWS CLI

Pour récupérer une liste de courses pour un canari spécifié

L'`get-canary-runs`exemple suivant récupère une liste de courses pour le canari nommé`demo_canary`.

```
aws synthetics get-canary-runs \
```

```
    --name demo_canary
```

Sortie :

```
{
    "CanaryRuns": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
            "Name": "demo_canary",
            "Status": {
                "State": "PASSED",
                "StateReason": "",
                "StateReasonCode": ""
            },
            "Timeline": {
                "Started": "2024-10-16T10:38:57.013000+05:30",
                "Completed": "2024-10-16T10:39:25.793000+05:30"
            },
            "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-abc-example1234/2024/10/15/13/50-39-690"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir GetCanaryRunsla section Référence des AWS CLI commandes.

`get-canary`

L'exemple de code suivant montre comment utiliser`get-canary`.

AWS CLI

Pour récupérer des informations complètes sur un canari

L'`get-canary`exemple suivant permet de récupérer des informations complètes sur le canari nommé`demo_canary`.

```
aws synthetics get-canary \
```

```
    --name demo_canary
```

Sortie :

```
{
    "Canary": {
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Name": "demo_canary",
        "Code": {
            "SourceLocationArn": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
            "Handler": "pageLoadBlueprint.handler"
        },
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/demo_canary_role",
        "Schedule": {
            "Expression": "rate(10 minutes)",
            "DurationInSeconds": 0
        },
        "RunConfig": {
            "TimeoutInSeconds": 300,
            "MemoryInMB": 1000,
            "ActiveTracing": false
        },
        "SuccessRetentionPeriodInDays": 31,
        "FailureRetentionPeriodInDays": 31,
        "Status": {
            "State": "RUNNING"
        },
        "Timeline": {
            "Created": "2024-10-15T18:55:15.168000+05:30",
            "LastModified": "2024-10-15T18:55:40.540000+05:30",
            "LastStarted": "2024-10-15T18:55:40.540000+05:30"
        },
        "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-a12-a123bc456789",
        "EngineArn": "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
        "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
        "Tags": {
            "blueprint": "heartbeat"
        }
    }
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir GetCanaryla section Référence des AWS CLI commandes.

`get-group`

L'exemple de code suivant montre comment utiliser`get-group`.

AWS CLI

Pour renvoyer des informations concernant un groupe

L'`get-group`exemple suivant renvoie des informations sur le groupe nommé`demo_group`.

```
aws synthetics get-group \
    --group-identifier demo_group
```

Sortie :

```
{
    "Group": {
        "Id": "example123",
        "Name": "demo_group",
        "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123",
        "Tags": {},
        "CreatedTime": "2024-10-15T14:47:23.811000+05:30",
        "LastModifiedTime": "2024-10-15T14:47:23.811000+05:30"
    }
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir GetGroupla section Référence des AWS CLI commandes.

`list-associated-groups`

L'exemple de code suivant montre comment utiliser`list-associated-groups`.

AWS CLI

Pour renvoyer la liste des groupes

L'`list-associated-groups`exemple suivant renvoie une liste des groupes associés au canari nommé`demo_canary`.

```
aws synthetics list-associated-groups \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

Sortie :

```
{
    "Groups": [
        {
            "Id": "example123",
            "Name": "demo_group",
            "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListAssociatedGroupsla section Référence des AWS CLI commandes.

`list-group-resources`

L'exemple de code suivant montre comment utiliser`list-group-resources`.

AWS CLI

Pour renvoyer une liste ARNs des canaris associés au groupe spécifié

L'`list-group-resources`exemple suivant renvoie une liste ARNs des canaris associés au groupe nommé`demo_group`.

```
aws synthetics list-group-resources \
    --group-identifier demo_group
```

Sortie :

```
{
    "Resources": [
```

```
        "arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary"
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListGroupResourcesla section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour renvoyer une liste de tous les groupes du compte

L'`list-groups`exemple suivant renvoie une liste de tous les groupes du compte.

```
aws synthetics list-groups
```

Sortie :

```
{
    "Groups": [
        {
            "Id": "example123",
            "Name": "demo_group",
            "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListGroupsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Exemple 1 : pour afficher les balises associées à un canari

L'`list-tags-for-resource`exemple suivant renvoie les balises associées à un canari nommé`demo_canary`.

```
aws synthetics list-tags-for-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

Sortie :

```
{
    "Tags": {
        "blueprint": "heartbeat"
    }
}
```

Exemple 2 : pour afficher les tags associés à un groupe

L'`list-tags-for-resource`exemple suivant renvoie les balises associées à un groupe nommé`demo_group`.

```
aws  synthetics list-tags-for-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123
```

Sortie :

```
{
    "Tags": {
        "team": "Devops"
    }
}
```

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`start-canary`

L'exemple de code suivant montre comment utiliser`start-canary`.

AWS CLI

Pour diriger un canari

L'`start-canary`exemple suivant exécute un canari nommé`demo_canary`.

```
aws synthetics start-canary \
    --name demo_canary
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir StartCanaryla section Référence des AWS CLI commandes.

`stop-canary`

L'exemple de code suivant montre comment utiliser`stop-canary`.

AWS CLI

Pour arrêter un canari

L'`stop-canary`exemple suivant arrête le canari nommé`demo_canary`.

```
aws synthetics stop-canary \
    --name demo_canary
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir StopCanaryla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Exemple 1 : Pour attribuer un tag au canari

L'`tag-resource`exemple suivant attribue une balise au canari nommé`demo_canary`.

```
aws synthetics tag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary \
    --tags blueprint=heartbeat
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour attribuer un tag au groupe

L'`tag-resource`exemple suivant attribue une balise au groupe nommé`demo_group`.

```
aws synthetics tag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123 \
    --tags team=Devops
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Exemple 1 : pour supprimer une étiquette du canari

L'`untag-resource`exemple suivant supprime une balise du canari nommé`demo_canary`.

```
aws synthetics untag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary \
```

```
    --tag-keys blueprint
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour supprimer un tag du groupe

L'`untag-resource`exemple suivant attribue une balise supprime une balise au groupe nommé`demo_group`.

```
aws synthetics untag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123 \
    --tag-keys team
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-canary`

L'exemple de code suivant montre comment utiliser`update-canary`.

AWS CLI

Pour mettre à jour un canari

L'`update-canary`exemple suivant met à jour la configuration d'un canari nommé`demo_canary`.

```
aws synthetics update-canary \
    --name demo_canary \
    --schedule Expression="rate(15 minutes)"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance synthétique (canaris) dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, voir UpdateCanaryla section Référence des AWS CLI commandes.

CodeArtifact exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CodeArtifact.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-external-connection`

L'exemple de code suivant montre comment utiliser`associate-external-connection`.

AWS CLI

Pour ajouter une connexion externe à un référentiel

L'`associate-external-connection`exemple suivant ajoute une connexion externe à npmjs.com à un référentiel nommé test-repo.

```
aws codeartifact associate-external-connection \
    --repository test-repo \
    --domain test-domain \
    --external-connection public:npmjs
```

Sortie :

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
```

```
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "upstreams": [],
        "externalConnections": [
            {
                "externalConnectionName": "public:npmjs",
                "packageFormat": "npm",
                "status": "AVAILABLE"
            }
        ]
    }
}
```

Pour plus d'informations, voir Ajouter une connexion externe dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir AssociateExternalConnectionla section Référence des AWS CLI commandes.

`copy-package-versions`

L'exemple de code suivant montre comment utiliser`copy-package-versions`.

AWS CLI

Pour copier des versions de packages d'un référentiel à un autre

Ce qui suit `copy-package-versions` déplace les versions 4.0.0 et 5.0.0 d'un package nommé test-package de my-repo vers test-repo.

```
aws codeartifact copy-package-versions \
    --domain test-domain \
    --source-repository my-repo \
    --destination-repository test-repo \
    --format npm \
    --package test-package \
    --versions '["4.0.0", "5.0.0"]'
```

Sortie :

```
{
```

```
    "format": "npm",
    "package": "test-package",
    "versions": [
        {
        "version": "5.0.0",
        "revision": "REVISION-1-SAMPLE-6C81EFF7DA55CC",
        "status": "Published"
        },
        {
        "version": "4.0.0",
        "revision": "REVISION-2-SAMPLE-55C752BEE772FC",
        "status": "Published"
        }
    ]
}
```

Pour plus d'informations, voir Copier des packages entre des référentiels dans le Guide de l'AWS CodeArtifact utilisateur.

- Pour plus de détails sur l'API, voir CopyPackageVersionsla section Référence des AWS CLI commandes.

`create-domain`

L'exemple de code suivant montre comment utiliser`create-domain`.

AWS CLI

Pour créer un domaine

L'`create-domain`exemple suivant crée un domaine nommé test-domain.

```
aws codeartifact create-domain \
    --domain test-domain
```

Sortie :

```
{
    "domain": {
        "name": "test-domain",
        "owner": "111122223333",
```

```
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",
        "status": "Active",
        "createdTime": "2020-10-20T13:16:48.559000-04:00",
        "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "repositoryCount": 0,
        "assetSizeBytes": 0
    }
}
```

Pour plus d'informations, consultez la section Création d'un domaine dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir CreateDomainla section Référence des AWS CLI commandes.

`create-repository`

L'exemple de code suivant montre comment utiliser`create-repository`.

AWS CLI

Pour créer un référentiel

L'`create-repository`exemple suivant crée un dépôt nommé test-repo dans un domaine nommé test-domain.

```
aws codeartifact create-repository \
    --domain test-domain \
    --domain-owner 111122223333 \
    --repository test-repo \
    --description "This is a test repository."
```

Sortie :

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
```

```
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "This is a test repository.",
        "upstreams": [],
        "externalConnections": []
    }
}
```

Pour plus d'informations, consultez la section Création d'un domaine dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir CreateRepositorylа section Référence des AWS CLI commandes.

`delete-domain-permissions-policy`

L'exemple de code suivant montre comment utiliser`delete-domain-permissions-policy`.

AWS CLI

Pour supprimer le document de politique d'autorisations d'un domaine

L'`delete-domain-permissions-policy`exemple suivant supprime la politique d'autorisation d'un domaine nommé test-domain.

```
aws codeartifact delete-domain-permissions-policy \
    --domain test-domain
```

Sortie :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BasicDomainPolicy",
            "Action": [
                "codeartifact:GetDomainPermissionsPolicy",
                "codeartifact:ListRepositoriesInDomain",
                "codeartifact:GetAuthorizationToken",
                "codeartifact:CreateRepository"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            }
        }
    ]
}
```

Pour plus d'informations, voir Supprimer une politique de domaine dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DeleteDomainPermissionsPolicyla section Référence des AWS CLI commandes.

`delete-domain`

L'exemple de code suivant montre comment utiliser`delete-domain`.

AWS CLI

Pour supprimer un domaine

L'`delete-domain`exemple suivant supprime un domaine nommé`test-domain`.

```
aws codeartifact delete-domain \
    --domain test-domain
```

Sortie :

```
{
    "domain": {
        "name": "test-domain",
        "owner": "417498243647",
        "arn": "arn:aws:codeartifact:us-west-2:417498243647:domain/test-domain",
        "status": "Deleted",
        "createdTime": "2020-10-20T13:16:48.559000-04:00",
        "encryptionKey": "arn:aws:kms:us-west-2:417498243647:key/c9fe2447-0795-4fda-
afbe-8464574ae162",
        "repositoryCount": 0,
        "assetSizeBytes": 0
    }
```

```
}
```

Pour plus d'informations, voir Supprimer un domaine dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DeleteDomainla section Référence des AWS CLI commandes.

`delete-package-versions`

L'exemple de code suivant montre comment utiliser`delete-package-versions`.

AWS CLI

Pour supprimer des versions de package

L'`delete-package-versions`exemple suivant supprime la version 4.0.0 d'un package nommé test-package.

```
aws codeartifact delete-package-versions \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0
```

Sortie :

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Deleted"
        }
    },
    "failedVersions": {}
}
```

Pour plus d'informations, voir Supprimer une version de package dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DeletePackageVersionsla section Référence des AWS CLI commandes.

`delete-repository-permissions-policy`

L'exemple de code suivant montre comment utiliser`delete-repository-permissions-policy`.

AWS CLI

Pour supprimer une politique d'autorisation d'un référentiel

L'`delete-repository-permissions-policy`exemple suivant supprime la politique d'autorisation d'un référentiel nommé test-repo.

```
aws codeartifact delete-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo
```

Sortie :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "codeartifact:DescribePackageVersion",
                "codeartifact:DescribeRepository",
                "codeartifact:GetPackageVersionReadme",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:ListPackages",
                "codeartifact:ListPackageVersions",
                "codeartifact:ListPackageVersionAssets",
                "codeartifact:ListPackageVersionDependencies",
                "codeartifact:ReadFromRepository"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

Pour plus d'informations, voir Supprimer une politique dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DeleteRepositoryPermissionsPolicyla section Référence des AWS CLI commandes.

`delete-repository`

L'exemple de code suivant montre comment utiliser`delete-repository`.

AWS CLI

Pour supprimer un référentiel

L'`delete-repository`exemple suivant supprime un référentiel nommé `test-repo` dans un domaine nommé`test-domain`.

```
aws codeartifact delete-repository \
    --domain test-domain \
    --repository test-repo
```

Sortie :

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "This is a test repository",
        "upstreams": [],
        "externalConnections": []
    }
}
```

Pour plus d'informations, voir Supprimer un référentiel dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DeleteRepositoryla section Référence des AWS CLI commandes.

`describe-domain`

L'exemple de code suivant montre comment utiliser`describe-domain`.

AWS CLI

Pour obtenir des informations sur un domaine

L'`describe-domain`exemple suivant renvoie un DomainDescription objet pour un domaine nommé test-domain.

```
aws codeartifact describe-domain \
    --domain test-domain
```

Sortie :

```
{
    "domain": {
        "name": "test-domain",
        "owner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",
        "status": "Active",
        "createdTime": "2020-10-20T13:16:48.559000-04:00",
        "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "repositoryCount": 2,
        "assetSizeBytes": 0,
        "s3BucketArn": "arn:aws:s3:::assets-111122223333-us-west-2"
    }
}
```

Pour plus d'informations, consultez la section Présentation du domaine dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DescribeDomainla section Référence des AWS CLI commandes.

`describe-repository`

L'exemple de code suivant montre comment utiliser`describe-repository`.

AWS CLI

Pour obtenir des informations sur un référentiel

L'`describe-repository`exemple suivant renvoie un RepositoryDescription objet pour un référentiel nommé test-repo.

```
aws codeartifact describe-repository \
    --domain test-domain \
    --repository test-repo
```

Sortie :

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "This is a test repository.",
        "upstreams": [],
        "externalConnections": []
    }
}
```

Pour plus d'informations, consultez la section Création d'un domaine dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DescribeRepositoryla section Référence des AWS CLI commandes.

`disassociate-external-connection`

L'exemple de code suivant montre comment utiliser`disassociate-external-connection`.

AWS CLI

Pour supprimer une connexion externe d'un référentiel

L'`disassociate-external-connection`exemple suivant supprime une connexion externe à npmjs.com depuis un dépôt nommé test-repo.

```
aws codeartifact disassociate-external-connection \
    --repository test-repo \
    --domain test-domain \
    --external-connection public:npmjs
```

Sortie :

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "upstreams": [],
        "externalConnections": []
    }
}
```

Pour plus d'informations, voir Supprimer une connexion externe dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DisassociateExternalConnectionla section Référence des AWS CLI commandes.

`dispose-package-versions`

L'exemple de code suivant montre comment utiliser`dispose-package-versions`.

AWS CLI

Pour supprimer les actifs d'une version de package et définir son statut sur Disposé

L'`dispose-package-versions`exemple suivant supprime les actifs du package de test version 4.0.0 et définit son statut sur Disposed.

```
aws codeartifact dispose-package-versions \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0
```

Sortie :

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Disposed"
        }
    },
    "failedVersions": {}
}
```

Pour plus d'informations, consultez la section Utilisation des packages CodeArtifact dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir DisposePackageVersionsla section Référence des AWS CLI commandes.

`get-authorization-token`

L'exemple de code suivant montre comment utiliser`get-authorization-token`.

AWS CLI

Pour obtenir un jeton d'autorisation

L'`get-authorization-token`exemple suivant récupère un jeton CodeArtifact d'autorisation.

```
aws codeartifact get-authorization-token \
    --domain test-domain \
    --query authorizationToken \
```

```
    --output text
```

Sortie :

```
This command will return the authorization token. You can store the output in an
 environment variable when calling the command.
```

Pour plus d'informations, voir Configurer pip sans la commande de connexion dans le Guide de l'AWS CodeArtifact utilisateur.

- Pour plus de détails sur l'API, voir GetAuthorizationTokenla section Référence des AWS CLI commandes.

`get-domain-permissions-policy`

L'exemple de code suivant montre comment utiliser`get-domain-permissions-policy`.

AWS CLI

Pour obtenir le document de politique d'autorisations pour un domaine

L'`get-domain-permissions-policy`exemple suivant montre comment associer la politique d'autorisation à un domaine nommé test-domain.

```
aws codeartifact get-domain-permissions-policy \
    --domain test-domain
```

Sortie :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BasicDomainPolicy",
            "Action": [
                "codeartifact:GetDomainPermissionsPolicy",
                "codeartifact:ListRepositoriesInDomain",
                "codeartifact:GetAuthorizationToken",
                "codeartifact:CreateRepository"
            ],
            "Effect": "Allow",
            "Resource": "*",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Lire une politique de domaine dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir GetDomainPermissionsPolicyla section Référence des AWS CLI commandes.

`get-package-version-asset`

L'exemple de code suivant montre comment utiliser`get-package-version-asset`.

AWS CLI

Pour obtenir un actif à partir d'une version de package

L'`get-package-version-asset`exemple suivant récupère l'`package.tgz`actif pour la version 4.0.0 d'un package npm nommé test-package.

```
aws codeartifact get-package-version-asset \
    --domain test-domain \
    --repository test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0 \
    --asset 'package.tgz' \
    outfileName
```

Sortie :

```
The output for this command will also store the raw asset in the file provided in
 place of outfileName.

{
    "assetName": "package.tgz",
    "packageVersion": "4.0.0",
    "packageVersionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="
```

```
}
```

Pour plus d'informations, voir Répertorier les actifs des versions du package dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir GetPackageVersionAssetla section Référence des AWS CLI commandes.

`get-package-version-readme`

L'exemple de code suivant montre comment utiliser`get-package-version-readme`.

AWS CLI

Pour obtenir le fichier readme d'une version de package

L'`get-package-version-readme`exemple suivant récupère le fichier readme pour la version 4.0.0 d'un package npm nommé test-package.

```
aws codeartifact get-package-version-readme \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

Sortie :

```
{
    "format": "npm",
    "package": "test-package",
    "version": "4.0.0",
    "readme": "<div align=\"center\">\n   <a href=\https://github.com/test-package/
testpack\"> ... more content ... \n",
    "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="
}
```

Pour plus d'informations, voir Afficher le fichier readme de la version du package dans le guide de l'AWS CodeArtifact utilisateur.

- Pour plus de détails sur l'API, voir GetPackageVersionReadmela section Référence des AWS CLI commandes.

`get-repository-endpoint`

L'exemple de code suivant montre comment utiliser`get-repository-endpoint`.

AWS CLI

Pour obtenir le point de terminaison URL d'un dépôt

L'`get-repository-endpoint`exemple suivant renvoie le point de terminaison npm pour le référentiel test-repo.

```
aws codeartifact get-repository-endpoint \
    --domain test-domain \
    --repository test-repo \
    --format npm
```

Sortie :

```
{
    "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-
west-2.amazonaws.com/npm/test-repo/"
}
```

Pour plus d'informations, voir Se connecter à un référentiel dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir GetRepositoryEndpointla section Référence des AWS CLI commandes.

`get-repository-permissions-policy`

L'exemple de code suivant montre comment utiliser`get-repository-permissions-policy`.

AWS CLI

Pour obtenir le document de politique d'autorisations pour un référentiel

L'`get-repository-permissions-policy`exemple suivant montre comment associer la politique d'autorisation à un référentiel nommé test-repo.

```
aws codeartifact get-repository-permissions-policy \
    --domain test-domain \
```

```
    --repository test-repo
```

Sortie :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "codeartifact:DescribePackageVersion",
                "codeartifact:DescribeRepository",
                "codeartifact:GetPackageVersionReadme",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:ListPackages",
                "codeartifact:ListPackageVersions",
                "codeartifact:ListPackageVersionAssets",
                "codeartifact:ListPackageVersionDependencies",
                "codeartifact:ReadFromRepository"
            ],
            "Resource": "*"
        }
    ]
}
```

Pour plus d'informations, voir Lire une politique dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir GetRepositoryPermissionsPolicyla section Référence des AWS CLI commandes.

`list-domains`

L'exemple de code suivant montre comment utiliser`list-domains`.

AWS CLI

Pour répertorier des domaines

L'`list-domains`exemple suivant renvoie un résumé de tous les domaines détenus par le AWS compte qui effectue l'appel.

```
aws codeartifact list-domains
```

Sortie :

```
{
    "domains": [
        {
            "name": "my-domain",
            "owner": "111122223333",
            "status": "Active",
            "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
        {
            "name": "test-domain",
            "owner": "111122223333",
            "status": "Active",
            "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des domaines CodeArtifact dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListDomainsla section Référence des AWS CLI commandes.

`list-package-version-assets`

L'exemple de code suivant montre comment utiliser`list-package-version-assets`.

AWS CLI

Pour afficher les ressources d'une version de package

L'`list-package-version-assets`exemple suivant récupère les actifs de la version 4.0.0 d'un package npm nommé test-package.

```
aws codeartifact list-package-version-assets \
    --domain test-domain \
    --repo test-repo \
```

```
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

Sortie :


```
{
    "format": "npm",
    "package": "test-package",
    "version": "4.0.0",
    "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
    "assets": [
        {
            "name": "package.tgz",
            "size": 316680,
            "hashes": {
                "MD5": "60078ec6d9e76b89fb55c860832742b2",
                "SHA-1": "b44a9b6297bcb698f1c51a3545a2b3b368d59c52",
                "SHA-256":
 "d2aa8c6afc3c8591765785a37d1c5acae482a8eb3ab9729ed28922692454f2e2",
                "SHA-512":
 "3e585d15c8a594e20d7de57b362ea81754c011acb2641a19f1b72c8531ea39825896bab344ae616a0a5a824cb9
            }
        }
    ]
}
```


Pour plus d'informations, voir Répertorier les actifs des versions du package dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListPackageVersionAssetsla section Référence des AWS CLI commandes.

`list-package-version-dependencies`

L'exemple de code suivant montre comment utiliser`list-package-version-dependencies`.

AWS CLI

Pour afficher les dépendances d'une version de package

L'`list-package-version-dependencies`exemple suivant récupère les dépendances pour la version 4.0.0 d'un package npm nommé test-package.

```
aws codeartifact list-package-version-dependencies \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

Sortie :

```
{
    "format": "npm",
    "package": "test-package",
    "version": "4.0.0",
    "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
    "dependencies": [
        {
            "namespace": "testns",
            "package": "testdep1",
            "dependencyType": "regular",
            "versionRequirement": "1.8.5"
        },
        {
            "namespace": "testns",
            "package": "testdep2",
            "dependencyType": "regular",
            "versionRequirement": "1.8.5"
        }
    ]
}
```

Pour plus d'informations, voir Afficher et mettre à jour les détails et les dépendances des versions du package dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListPackageVersionDependencieslа section Référence des AWS CLI commandes.

`list-package-versions`

L'exemple de code suivant montre comment utiliser`list-package-versions`.

AWS CLI

Pour répertorier les versions d'un package

L'`list-package-versions`exemple suivant renvoie une liste des versions de package pour un package nommé`kind-of`.

```
aws codeartifact list-package-versions \
    --package kind-of \
    --domain test-domain \
    --repository test-repo \
    --format npm
```

Sortie :

```
{
    "defaultDisplayVersion": "1.0.1",
    "format": "npm",
    "package": "kind-of",
    "versions": [
        {
            "version": "1.0.1",
            "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC",
            "status": "Published"
        },
        {
            "version": "1.0.0",
            "revision": "REVISION-SAMPLE-2-C752BEEF6D2CFC",
            "status": "Published"
        },
        {
            "version": "0.1.2",
            "revision": "REVISION-SAMPLE-3-654S65A5C5E1FC",
            "status": "Published"
        },
        {
            "version": "0.1.1",
            "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC"",
            "status": "Published"
        },
        {
            "version": "0.1.0",
            "revision": "REVISION-SAMPLE-4-AF669139B772FC",
```

```
            "status": "Published"
        }
    ]
}
```

Pour plus d'informations, voir Répertorier les versions des packages dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListPackageVersionsla section Référence des AWS CLI commandes.

`list-packages`

L'exemple de code suivant montre comment utiliser`list-packages`.

AWS CLI

Pour répertorier les packages dans un référentiel

L'`list-packages`exemple suivant répertorie les packages d'un référentiel nommé `test-repo` dans un domaine nommé`test-domain`.

```
aws codeartifact list-packages \
    --domain test-domain \
    --repository test-repo
```

Sortie :

```
{
    "packages": [
        {
            "format": "npm",
            "package": "lodash"
        }
        {
            "format": "python",
            "package": "test-package"
        }
    ]
}
```

Pour plus d'informations, voir Répertorier les noms de packages dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListPackagesla section Référence des AWS CLI commandes.

`list-repositories-in-domain`

L'exemple de code suivant montre comment utiliser`list-repositories-in-domain`.

AWS CLI

Pour répertorier les référentiels d'un domaine

L'`list-repositories-in-domain`exemple suivant renvoie un résumé de tous les référentiels du domaine de test.

```
aws codeartifact list-repositories-in-domain \
    --domain test-domain
```

Sortie :

```
{
    "repositories": [
        {
            "name": "test-repo",
            "administratorAccount": "111122223333",
            "domainName": "test-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo",
            "description": "This is a test repository."
        },
        {
            "name": "test-repo2",
            "administratorAccount": "111122223333",
            "domainName": "test-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo2",
            "description": "This is a test repository."
        }
    ]
```

```
}
```

Pour plus d'informations, voir Répertorier les référentiels dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListRepositoriesInDomainla section Référence des AWS CLI commandes.

`list-repositories`

L'exemple de code suivant montre comment utiliser`list-repositories`.

AWS CLI

Pour répertorier les référentiels

L'`list-repositories`exemple suivant renvoie un résumé de tous les référentiels du domaine appartenant au AWS compte qui effectue l'appel.

```
aws codeartifact list-repositories
```

Sortie :

```
{
    "repositories": [
        {
            "name": "npm-store",
            "administratorAccount": "111122223333",
            "domainName": "my-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-
domain/npm-store",
            "description": "Provides npm artifacts from npm, Inc."
        },
        {
            "name": "target-repo",
            "administratorAccount": "111122223333",
            "domainName": "my-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-
domain/target-repo",
            "description": "test target repo"
        },
```

```
        {
            "name": "test-repo2",
            "administratorAccount": "111122223333",
            "domainName": "test-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo2",
            "description": "This is a test repository."
        }
    ]
}
```

Pour plus d'informations, voir Répertorier les référentiels dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir ListRepositoriesla section Référence des AWS CLI commandes.

`login`

L'exemple de code suivant montre comment utiliser`login`.

AWS CLI

Pour configurer l'authentification auprès de votre référentiel à l'aide de la commande de connexion

L'`login`exemple suivant configure le gestionnaire de packages npm avec un référentiel nommé test-repo dans un domaine nommé test-domain.

```
aws codeartifact login \
    --domain test-domain \
    --repository test-repo \
    --tool npm
```

Sortie :

```
Successfully configured npm to use AWS CodeArtifact repository https://test-
domain-111122223333.d.codeartifact.us-west-2.amazonaws.com/npm/test-repo/
Login expires in 12 hours at 2020-11-12 01:53:16-05:00
```

Pour plus d'informations, consultez la section Getting started with the AWS CLI dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir Login in AWS CLI Command Reference.

`put-domain-permissions-policy`

L'exemple de code suivant montre comment utiliser`put-domain-permissions-policy`.

AWS CLI

Pour associer une politique d'autorisation à un domaine

L'`put-domain-permissions-policy`exemple suivant attache une politique d'autorisation définie dans le fichier policy.json à un domaine nommé test-domain.

```
aws codeartifact put-domain-permissions-policy \
    --domain test-domain \
    --policy-document file://PATH/TO/policy.json
```

Sortie :

```
{
    "policy": {
        "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-
domain",
        "document": "{ ...policy document content...}",
        "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxx="
    }
}
```

Pour plus d'informations, voir Définir une politique de domaine dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir PutDomainPermissionsPolicyla section Référence des AWS CLI commandes.

`put-repository-permissions-policy`

L'exemple de code suivant montre comment utiliser`put-repository-permissions-policy`.

AWS CLI

Pour associer une politique d'autorisation à un référentiel

L'`put-repository-permissions-policy`exemple suivant joint une politique d'autorisation définie dans le fichier policy.json à un référentiel nommé test-repo.

```
aws codeartifact put-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo \
    --policy-document file://PATH/TO/policy.json
```

Sortie :

```
{
    "policy": {
        "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-
domain/test-repo",
        "document": "{ ...policy document content...}",
        "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxx="
    }
}
```

Pour plus d'informations, voir Définir une politique dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir PutRepositoryPermissionsPolicyla section Référence des AWS CLI commandes.

`update-package-versions-status`

L'exemple de code suivant montre comment utiliser`update-package-versions-status`.

AWS CLI

Pour mettre à jour l'état de la version du package

L'`update-package-versions-status`exemple suivant met à jour le statut de la version 4.0.0 du package de test sur Archivé.

```
aws codeartifact update-package-versions-status \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0 \
```

```
    --target-status Archived
```

Sortie :

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Archived"
        }
    },
    "failedVersions": {}
}
```

Pour plus d'informations, consultez la section État de la version du package de mise à jour dans le guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir UpdatePackageVersionsStatusla section Référence des AWS CLI commandes.

`update-repository`

L'exemple de code suivant montre comment utiliser`update-repository`.

AWS CLI

Pour mettre à jour un référentiel

L'`update-repository`exemple suivant met à jour la description d'un dépôt nommé test-repo dans un domaine nommé test-domain en « ceci est une description mise à jour ».

```
aws codeartifact update-repository \
    --domain test-domain \
    --repository test-repo \
    --description "this is an updated description"
```

Sortie :

```
{
    "repository": {
        "name": "test-repo",
```

```
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "this is an updated description",
        "upstreams": [],
        "externalConnections": []
    }
}
```

Pour plus d'informations, voir Afficher ou modifier une configuration de référentiel dans le Guide de AWS CodeArtifact l'utilisateur.

- Pour plus de détails sur l'API, voir UpdateRepositoryla section Référence des AWS CLI commandes.

CodeBuild exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CodeBuild.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-delete-builds`

L'exemple de code suivant montre comment utiliser`batch-delete-builds`.

AWS CLI

Pour supprimer des versions intégrées AWS CodeBuild.

L'`batch-delete-builds`exemple suivant supprime les versions intégrées CodeBuild avec le paramètre spécifié IDs.

```
aws codebuild batch-delete-builds --ids my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE
```

Sortie :

```
{
    "buildsNotDeleted": [
        {
            "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE",
            "statusCode": "BUILD_IN_PROGRESS"
        }
    ],
    "buildsDeleted": [
        "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE"
    ]
}
```

Pour plus d'informations, consultez la section Delete Builds (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchDeleteBuildsà la section Référence des AWS CLI commandes.

`batch-get-build-batches`

L'exemple de code suivant montre comment utiliser`batch-get-build-batches`.

AWS CLI

Pour afficher les détails des intégrations AWS CodeBuild.

L'`batch-get-build-batches`exemple suivant obtient des informations sur les lots de construction CodeBuild avec le paramètre spécifié IDs.

```
aws codebuild batch-get-build-batches \
    --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```

Sortie :


```
{
    "buildBatches": [
        {
            "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build-batch/codebuild-
demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "startTime": "2020-11-03T21:52:20.775000+00:00",
            "endTime": "2020-11-03T21:56:59.784000+00:00",
            "currentPhase": "SUCCEEDED",
            "buildBatchStatus": "SUCCEEDED",
            "resolvedSourceVersion": "0a6546f68309560d08a310daac92314c4d378f6b",
            "projectName": "codebuild-demo-project",
            "phases": [
                {
                    "phaseType": "SUBMITTED",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:52:20.775000+00:00",
                    "endTime": "2020-11-03T21:52:20.976000+00:00",
                    "durationInSeconds": 0
                },
                {
                    "phaseType": "DOWNLOAD_BATCHSPEC",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:52:20.976000+00:00",
                    "endTime": "2020-11-03T21:52:57.401000+00:00",
                    "durationInSeconds": 36
                },
                {
                    "phaseType": "IN_PROGRESS",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:52:57.401000+00:00",
                    "endTime": "2020-11-03T21:56:59.751000+00:00",
                    "durationInSeconds": 242
                },
                {
                    "phaseType": "COMBINE_ARTIFACTS",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:56:59.751000+00:00",
                    "endTime": "2020-11-03T21:56:59.784000+00:00",
                    "durationInSeconds": 0
                },
                {
```

```
                "phaseType": "SUCCEEDED",
                "startTime": "2020-11-03T21:56:59.784000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "https://github.com/my-repo/codebuild-demo-project.git",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "complete": true,
        "initiator": "Strohm",
```

```
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "buildBatchNumber": 6,
            "buildBatchConfig": {
                "serviceRole": "arn:aws:iam::123456789012:role/service-role/
codebuild-demo-project",
                "restrictions": {
                    "maximumBuildsAllowed": 100
                },
                "timeoutInMins": 480
            },
            "buildGroups": [
                {
                    "identifier": "DOWNLOAD_SOURCE",
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:379737d8-bc35-48ec-97fd-776d27545315",
                        "requestedOn": "2020-11-03T21:52:21.394000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "DOWNLOAD_SOURCE"
                        },
                        "secondaryArtifacts": []
                    }
                },
                {
                    "identifier": "linux_small",
                    "dependsOn": [],
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:dd785171-ed84-4bb6-8ede-ceeb86e54bdb",
                        "requestedOn": "2020-11-03T21:52:57.604000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "linux_small"
                        },
                        "secondaryArtifacts": []
                    }
                },
                {
                    "identifier": "linux_medium",
```

```
                    "dependsOn": [
                        "linux_small"
                    ],
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:97cf7bd4-5313-4786-8243-4aef350a1267",
                        "requestedOn": "2020-11-03T21:54:18.474000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "linux_medium"
                        },
                        "secondaryArtifacts": []
                    }
                },
                {
                    "identifier": "linux_large",
                    "dependsOn": [
                        "linux_medium"
                    ],
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:60a194cd-0d03-4337-9db1-d41476a17d27",
                        "requestedOn": "2020-11-03T21:55:39.203000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "linux_large"
                        },
                        "secondaryArtifacts": []
                    }
                }
            ]
        }
    ],
    "buildBatchesNotFound": []
}
```

Pour plus d'informations, consultez Batch builds in AWS CodeBuild < https:// docs.aws.amazon.com/codebuild/ latest/userguide/batch -build.html>) __ dans le guide de l'utilisateur.AWS CodeBuild

- Pour plus de détails sur l'API, reportez-vous BatchGetBuildBatchesà la section Référence des AWS CLI commandes.

`batch-get-builds`

L'exemple de code suivant montre comment utiliser`batch-get-builds`.

AWS CLI

Pour afficher les détails des intégrations AWS CodeBuild.

L'`batch-get-builds`exemple suivant obtient des informations sur les intégrations CodeBuild avec le paramètre spécifié IDs.

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-
ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

Sortie :

```
{
    "buildsNotFound": [],
    "builds": [
        {
            "artifacts": {
                "md5sum": "0e95edf915048a0c22efe6d139fff837",
                "location": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/BuildArtif/6DJsqQa",
                "encryptionDisabled": false,
                "sha256sum":
 "cfa0df33a090966a737f64ae4fe498969fdc842a0c9aec540bf93c37ac0d05a2"
            },
            "logs": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
                },
                "s3Logs": {
                    "status": "DISABLED"
                },
                "streamName": "46472baf-8f6b-43c2-9255-b3b963af2732",
                "groupName": "/aws/codebuild/codebuild-demo-project",
```

```
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=46472baf-8f6b-43c2-9255-b3b963af2732"
            },
            "timeoutInMinutes": 60,
            "environment": {
                "privilegedMode": false,
                "computeType": "BUILD_GENERAL1_MEDIUM",
                "image": "aws/codebuild/windows-base:1.0",
                "environmentVariables": [],
                "type": "WINDOWS_CONTAINER"
            },
            "projectName": "codebuild-demo-project",
            "buildComplete": true,
            "source": {
                "gitCloneDepth": 1,
                "insecureSsl": false,
                "type": "CODEPIPELINE"
            },
            "buildStatus": "SUCCEEDED",
            "secondaryArtifacts": [],
            "phases": [
                {
                    "durationInSeconds": 0,
                    "startTime": 1548717462.122,
                    "phaseType": "SUBMITTED",
                    "endTime": 1548717462.484,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "startTime": 1548717462.484,
                    "phaseType": "QUEUED",
                    "endTime": 1548717462.775,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 34,
                    "endTime": 1548717496.909,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
```

```
                ],
                "startTime": 1548717462.775,
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 15,
                "endTime": 1548717512.555,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717496.909,
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548717512.734,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717512.555,
                "phaseType": "INSTALL",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548717512.924,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717512.734,
                "phaseType": "PRE_BUILD",
                "phaseStatus": "SUCCEEDED"
            },
```

```
                {
                    "durationInSeconds": 9,
                    "endTime": 1548717522.254,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717512.924,
                    "phaseType": "BUILD",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 3,
                    "endTime": 1548717525.498,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717522.254,
                    "phaseType": "POST_BUILD",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 9,
                    "endTime": 1548717534.646,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717525.498,
                    "phaseType": "UPLOAD_ARTIFACTS",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 2,
                    "endTime": 1548717536.846,
                    "contexts": [
                        {
```

```
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717534.646,
                    "phaseType": "FINALIZING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "startTime": 1548717536.846,
                    "phaseType": "COMPLETED"
                }
            ],
            "startTime": 1548717462.122,
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "initiator": "codepipeline/CodeBuild-Pipeline",
            "secondarySources": [],
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
            "currentPhase": "COMPLETED",
            "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "cache": {
                "type": "NO_CACHE"
            },
            "sourceVersion": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/SourceArti/1TspnN3.zip",
            "endTime": 1548717536.846,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "queuedTimeoutInMinutes": 480,
            "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
        },
        {
            "artifacts": {
                "md5sum": "",
                "overrideArtifactName": false,
                "location": "arn:aws:s3:::my-artifacts/codebuild-demo-project",
                "encryptionDisabled": false,
                "sha256sum": ""
            },
            "logs": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
                },
```

```
                "s3Logs": {
                    "status": "DISABLED"
                },
                "streamName": "4dea3ca4-20ec-4898-b22a-a9eb9292775d",
                "groupName": "/aws/codebuild/codebuild-demo-project",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=4dea3ca4-20ec-4898-b22a-a9eb9292775d"
            },
            "timeoutInMinutes": 60,
            "environment": {
                "privilegedMode": false,
                "computeType": "BUILD_GENERAL1_MEDIUM",
                "image": "aws/codebuild/windows-base:1.0",
                "environmentVariables": [],
                "type": "WINDOWS_CONTAINER"
            },
            "projectName": "codebuild-demo-project",
            "buildComplete": true,
            "source": {
                "gitCloneDepth": 1,
                "location": "https://github.com/my-repo/codebuild-demo-project.git",
                "insecureSsl": false,
                "reportBuildStatus": false,
                "type": "GITHUB"
            },
            "buildStatus": "SUCCEEDED",
            "secondaryArtifacts": [],
            "phases": [
                {
                    "durationInSeconds": 0,
                    "startTime": 1548716241.89,
                    "phaseType": "SUBMITTED",
                    "endTime": 1548716242.241,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "startTime": 1548716242.241,
                    "phaseType": "QUEUED",
                    "endTime": 1548716242.536,
                    "phaseStatus": "SUCCEEDED"
                },
                {
```

```
                    "durationInSeconds": 33,
                    "endTime": 1548716276.171,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716242.536,
                    "phaseType": "PROVISIONING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 15,
                    "endTime": 1548716291.809,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716276.171,
                    "phaseType": "DOWNLOAD_SOURCE",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "endTime": 1548716291.993,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716291.809,
                    "phaseType": "INSTALL",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "endTime": 1548716292.191,
                    "contexts": [
                        {
                            "statusCode": "",
```

```
                        "message": ""
                    }
                ],
                "startTime": 1548716291.993,
                "phaseType": "PRE_BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 9,
                "endTime": 1548716301.622,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716292.191,
                "phaseType": "BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 3,
                "endTime": 1548716304.783,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716301.622,
                "phaseType": "POST_BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 8,
                "endTime": 1548716313.775,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716304.783,
                "phaseType": "UPLOAD_ARTIFACTS",
```

```
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 2,
                    "endTime": 1548716315.935,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716313.775,
                    "phaseType": "FINALIZING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "startTime": 1548716315.935,
                    "phaseType": "COMPLETED"
                }
            ],
            "startTime": 1548716241.89,
            "secondarySourceVersions": [],
            "initiator": "my-codebuild-project",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
            "currentPhase": "COMPLETED",
            "id": "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
            "cache": {
                "type": "NO_CACHE"
            },
            "endTime": 1548716315.935,
            "secondarySources": [],
            "queuedTimeoutInMinutes": 480,
            "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
        }
    ]
}
```


Pour plus d'informations, consultez la section Afficher les détails de la version (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchGetBuildsà la section Référence des AWS CLI commandes.

`batch-get-projects`

L'exemple de code suivant montre comment utiliser`batch-get-projects`.

AWS CLI

Pour obtenir une liste des noms de projets de AWS CodeBuild construction.

L'`batch-get-projects`exemple suivant obtient une liste de projets de CodeBuild construction spécifiés par leur nom.

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2 my-other-demo-project
```

Dans le résultat suivant, le `projectsNotFound` tableau répertorie tous les noms de projets de construction qui ont été spécifiés, mais qui sont introuvables. Le tableau `projects` répertorie les détails pour chaque projet de génération pour lequel des informations ont été trouvées.

```
{
    "projectsNotFound": [],
    "projects": [
        {
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "name": "codebuild-demo-project2",
            "queuedTimeoutInMinutes": 480,
            "timeoutInMinutes": 60,
            "source": {
                "buildspec": "version: 0.2\n\n#env:\n  #variables:\n     # key: \"value\"\n     # key: \"value\"\n  #parameter-store:\n     # key: \"value\"\n  # key:\"value\"\n\nphases:\n  #install:\n    #commands:\n      # - command\n  # - command\n  #pre_build:\n    #commands:\n      # - command\n      # - command\n  build:\n    commands:\n      # - command\n      # - command\n  #post_build:\n    #commands:\n      # - command\n      # - command\n#artifacts:\n  #files:\n    # - location\n    # - location\n  #name: $(date +%Y-%m-%d)\n  #discard-paths: yes\n  #base-directory: location\n#cache:\n  #paths:\n    # - paths",
                "type": "NO_SOURCE",
                "insecureSsl": false,
                "gitCloneDepth": 1
            },
```

```
            "artifacts": {
                "type": "NO_ARTIFACTS"
            },
            "badge": {
                "badgeEnabled": false
            },
            "lastModified": 1540588091.108,
            "created": 1540588091.108,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:project/test-for-
sample",
            "secondarySources": [],
            "secondaryArtifacts": [],
            "cache": {
                "type": "NO_CACHE"
            },
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-test-
role",
            "environment": {
                "image": "aws/codebuild/java:openjdk-8",
                "privilegedMode": true,
                "type": "LINUX_CONTAINER",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": []
            },
            "tags": []
        },
        {
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "name": "my-other-demo-project",
            "queuedTimeoutInMinutes": 480,
            "timeoutInMinutes": 60,
            "source": {
                "location": "https://github.com/iversonic/codedeploy-sample.git",
                "reportBuildStatus": false,
                "buildspec": "buildspec.yml",
                "insecureSsl": false,
                "gitCloneDepth": 1,
                "type": "GITHUB",
                "auth": {
                    "type": "OAUTH"
                }
            },
            "artifacts": {
                "type": "NO_ARTIFACTS"
```

```
            },
            "badge": {
                "badgeEnabled": false
            },
            "lastModified": 1523401711.73,
            "created": 1523401711.73,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:project/Project2",
            "cache": {
                "type": "NO_CACHE"
            },
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/codebuild-
Project2-service-role",
            "environment": {
                "image": "aws/codebuild/nodejs:4.4.7",
                "privilegedMode": false,
                "type": "LINUX_CONTAINER",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": []
            },
            "tags": []
        }
    ]
}
```

Pour plus d'informations, voir Afficher les détails d'un projet de construction (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchGetProjectsà la section Référence des AWS CLI commandes.

`batch-get-report-groups`

L'exemple de code suivant montre comment utiliser`batch-get-report-groups`.

AWS CLI

Pour obtenir des informations sur un ou plusieurs groupes de rapports dans AWS CodeBuild.

L'`batch-get-report-groups`exemple suivant extrait des informations sur le groupe de rapports avec l'ARN spécifié.

```
aws codebuild batch-get-report-groups \
```

```
    --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>
```

Sortie :

```
{
    "reportGroups": [
        {
            "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-
group-name>",
            "name": "report-group-name",
            "type": "TEST",
            "exportConfig": {
                "exportConfigType": "NO_EXPORT"
            },
            "created": "2020-10-01T18:04:08.466000+00:00",
            "lastModified": "2020-10-01T18:04:08.466000+00:00",
            "tags": []
        }
    ],
    "reportGroupsNotFound": []
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchGetReportGroupsà la section Référence des AWS CLI commandes.

`batch-get-reports`

L'exemple de code suivant montre comment utiliser`batch-get-reports`.

AWS CLI

Pour obtenir des informations sur un ou plusieurs rapports dans AWS CodeBuild.

L'`batch-get-reports`exemple suivant permet de récupérer des informations sur les rapports contenant le paramètre spécifié ARNs.

```
aws codebuild batch-get-reports \
```

```
    --report-arns arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 1 ID> arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 2 ID>
```

Sortie :

```
{
    "reports": [
        {
            "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 1 ID>",
            "type": "TEST",
            "name": "<report-group-name>",
            "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>",
            "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-
reports:<ID>",
            "status": "FAILED",
            "created": "2020-10-01T11:25:22.531000-07:00",
            "expired": "2020-10-31T11:25:22-07:00",
            "exportConfig": {
                "exportConfigType": "NO_EXPORT"
            },
            "truncated": false,
            "testSummary": {
                "total": 28,
                "statusCounts": {
                    "ERROR": 5,
                    "FAILED": 1,
                    "SKIPPED": 4,
                    "SUCCEEDED": 18,
                    "UNKNOWN": 0
                },
                "durationInNanoSeconds": 94000000
            }
        },
        {
            "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 2 ID>",
            "type": "TEST",
            "name": "<report-group-name>",
            "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>",
```

```
            "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-
reports:<ID>",
            "status": "FAILED",
            "created": "2020-10-01T11:13:05.816000-07:00",
            "expired": "2020-10-31T11:13:05-07:00",
            "exportConfig": {
                "exportConfigType": "NO_EXPORT"
            },
            "truncated": false,
            "testSummary": {
                "total": 28,
                "statusCounts": {
                    "ERROR": 5,
                    "FAILED": 1,
                    "SKIPPED": 4,
                    "SUCCEEDED": 18,
                    "UNKNOWN": 0
                },
                "durationInNanoSeconds": 94000000
            }
        }
    ],
    "reportsNotFound": []
}
```

Pour plus d'informations, consultez la section Utilisation des rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchGetReportsà la section Référence des AWS CLI commandes.

`create-project`

L'exemple de code suivant montre comment utiliser`create-project`.

AWS CLI

Exemple 1 : pour créer un projet AWS CodeBuild de construction

L'`create-project`exemple suivant crée un projet de CodeBuild construction à l'aide des fichiers source d'un compartiment S3.

```
aws codebuild create-project \
```

```
    --name "my-demo-project" \
    --source "{\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source.zip\"}" \
    --artifacts {"\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-
output-bucket\""} \
    --environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/
standard:1.0\",\"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
    --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```

Sortie :

```
{
    "project": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
        "name": "my-cli-demo-project",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "lastModified": 1556839783.274,
        "badge": {
            "badgeEnabled": false
        },
        "queuedTimeoutInMinutes": 480,
        "environment": {
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "type": "LINUX_CONTAINER",
            "imagePullCredentialsType": "CODEBUILD",
            "privilegedMode": false,
            "environmentVariables": []
        },
        "artifacts": {
            "location": "codebuild-us-west-2-123456789012-output-bucket",
            "name": "my-cli-demo-project",
            "namespaceType": "NONE",
            "type": "S3",
            "packaging": "NONE",
            "encryptionDisabled": false
        },
        "source": {
            "type": "S3",
```

```
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
            "insecureSsl": false
        },
        "timeoutInMinutes": 60,
        "cache": {
            "type": "NO_CACHE"
        },
        "created": 1556839783.274
    }
}
```

Exemple 2 : pour créer un projet de AWS CodeBuild construction à l'aide d'un fichier d'entrée JSON pour les paramètres

L'`create-project`exemple suivant crée un projet de CodeBuild construction en transmettant tous les paramètres requis dans un fichier d'entrée JSON. Créez le modèle de fichier d'entrée en exécutant la commande avec uniquement le`--generate-cli-skeleton parameter`.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Le fichier JSON d'entrée `create-project.json` contient le contenu suivant :

```
{
    "name": "codebuild-demo-project",
    "source": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "artifacts": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-output-bucket"
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:1.0",
        "computeType": "BUILD_GENERAL1_SMALL"
    },
    "serviceRole": "serviceIAMRole"
}
```

Sortie :

```
{
    "project": {
        "name": "codebuild-demo-project",
        "serviceRole": "serviceIAMRole",
        "tags": [],
        "artifacts": {
            "packaging": "NONE",
            "type": "S3",
            "location": "codebuild-region-ID-account-ID-output-bucket",
            "name": "message-util.zip"
        },
        "lastModified": 1472661575.244,
        "timeoutInMinutes": 60,
        "created": 1472661575.244,
        "environment": {
            "computeType": "BUILD_GENERAL1_SMALL",
            "image": "aws/codebuild/standard:1.0",
            "type": "LINUX_CONTAINER",
            "environmentVariables": []
        },
        "source": {
            "type": "S3",
            "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
        },
        "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
        "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
    }
}
```

Pour plus d'informations, voir Create a Build Project (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateProjectà la section Référence des AWS CLI commandes.

`create-report-group`

L'exemple de code suivant montre comment utiliser`create-report-group`.

AWS CLI

Pour créer un groupe de rapports dans AWS CodeBuild.

L'`create-report-group`exemple suivant crée un nouveau groupe de rapports.

```
aws codebuild create-report-group \
    --cli-input-json file://create-report-group-source.json
```

Contenu du create-report-group-source fichier .json :

```
{
    "name": "cli-created-report-group",
    "type": "TEST",
    "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
            "bucket": "amzn-s3-demo-bucket",
            "path": "",
            "packaging": "ZIP",
            "encryptionDisabled": true
        }
    }
}
```

Sortie :

```
{
    "reportGroup": {
        "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-
report-group",
        "name": "cli-created-report-group",
        "type": "TEST",
        "exportConfig": {
            "exportConfigType": "S3",
            "s3Destination": {
                "bucket": "amzn-s3-demo-bucket",
                "path": "",
                "packaging": "ZIP",
                "encryptionDisabled": true
            }
        },
        "created": 1602020026.775,
```

```
        "lastModified": 1602020026.775
    }
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateReportGroupà la section Référence des AWS CLI commandes.

`create-webhook`

L'exemple de code suivant montre comment utiliser`create-webhook`.

AWS CLI

Pour créer des filtres de webhook pour un projet AWS CodeBuild

L'`create-webhook`exemple suivant crée un webhook pour un CodeBuild projet nommé `my-project` comportant deux groupes de filtres. Le premier groupe de filtres spécifie les demandes d'extraction créées, mises à jour ou rouvertes sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/master$` et des références principales qui correspondent à `^refs/heads/myBranch$`. Le deuxième groupe de filtres spécifie les requêtes push sur les branches dont les noms de référence Git ne correspondent pas à l'expression régulière`^refs/heads/myBranch$`.

```
aws codebuild create-webhook \
    --project-name my-project \
    --filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"PULL_REQUEST_CREATED,
 PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED\"},{\"type\":\"HEAD_REF\",\"pattern
\":\"^refs/heads/myBranch$\",\"excludeMatchedPattern\":true},{\"type\":\"BASE_REF
\",\"pattern\":\"^refs/heads/master$\",\"excludeMatchedPattern\":true}],[{\"type\":
\"EVENT\",\"pattern\":\"PUSH\"},{\"type\":\"HEAD_REF\",\"pattern\":\"^refs/heads/
myBranch$\",\"excludeMatchedPattern\":true}]]"
```

Sortie :

```
{
    "webhook": {
        "payloadUrl": "https://codebuild.us-west-2.amazonaws.com/webhooks?
t=eyJlbmNyeXB0ZWREYXRhIjoiVVl5MGtoeGRwSzZFRXl2Wnh4bld1Z0tKZ291TVpQNEtFamQ3RDlDYWpRaGIreVFrdm
```

```
        "url": "https://api.github.com/repos/iversonic/codedeploy-sample/
hooks/105190656",
        "lastModifiedSecret": 1556311319.069,
        "filterGroups": [
            [
                {
                    "type": "EVENT",
                    "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED",
                    "excludeMatchedPattern": false
                },
                {
                    "type": "HEAD_REF",
                    "pattern": "refs/heads/myBranch$",
                    "excludeMatchedPattern": true
                },
                {
                    "type": "BASE_REF",
                    "pattern": "refs/heads/master$",
                    "excludeMatchedPattern": true
                }
            ],
            [
                {
                    "type": "EVENT",
                    "pattern": "PUSH",
                    "excludeMatchedPattern": false
                },
                {
                    "type": "HEAD_REF",
                    "pattern": "refs/heads/myBranch$",
                    "excludeMatchedPattern": true
                }
            ]
        ]
    }
}
```

Pour plus d'informations, consultez Filter GitHub Webhook Events (SDK) dans le guide de l'AWS CodeBuild utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateWebhookà la section Référence des AWS CLI commandes.

`delete-build-batch`

L'exemple de code suivant montre comment utiliser`delete-build-batch`.

AWS CLI

Pour supprimer un batch intégré AWS CodeBuild.

L'`delete-build-batch`exemple suivant supprime la version par lots spécifiée.

```
aws codebuild delete-build-batch \
    --id <project-name>:<batch-ID>
```

Sortie :

```
{
    "statusCode": "BATCH_DELETED",
    "buildsDeleted": [
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-
ID>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-
ID>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-
ID>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>"
    ],
    "buildsNotDeleted": []
}
```

Pour plus d'informations, consultez la section Batch builds in AWS CodeBuild dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteBuildBatchà la section Référence des AWS CLI commandes.

`delete-project`

L'exemple de code suivant montre comment utiliser`delete-project`.

AWS CLI

Pour supprimer un projet AWS CodeBuild de construction

L'`delete-project`exemple suivant supprime le projet de CodeBuild construction spécifié.

```
aws codebuild delete-project --name my-project
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer un projet de construction (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteProjectà la section Référence des AWS CLI commandes.

`delete-report-group`

L'exemple de code suivant montre comment utiliser`delete-report-group`.

AWS CLI

Pour supprimer un rapport, des groupes dans AWS CodeBuild.

L'`delete-report-group`exemple suivant supprime le groupe de rapports avec l'ARN spécifié.

```
aws codebuild delete-report-group \
    --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteReportGroupà la section Référence des AWS CLI commandes.

`delete-report`

L'exemple de code suivant montre comment utiliser`delete-report`.

AWS CLI

Pour supprimer un rapport dans AWS CodeBuild.

L'`delete-report`exemple suivant supprime le rapport spécifié.

```
aws codebuild delete-report \
    --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteReportà la section Référence des AWS CLI commandes.

`delete-source-credentials`

L'exemple de code suivant montre comment utiliser`delete-source-credentials`.

AWS CLI

Pour vous déconnecter d'un fournisseur source et supprimer ses jetons d'accès.

L'`delete-source-credentials`exemple suivant se déconnecte d'un fournisseur source et supprime ses jetons. L'ARN des informations d'identification de source utilisées pour se connecter au fournisseur de source détermine les informations d'identification de la source.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Sortie :

```
{
    "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"
}
```

Pour plus d'informations, voir Connect Source Providers with Access Tokens (CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteSourceCredentialsà la section Référence des AWS CLI commandes.

`delete-webhook`

L'exemple de code suivant montre comment utiliser`delete-webhook`.

AWS CLI

Pour supprimer un filtre Webhook d'un projet AWS CodeBuild

L'`delete-webhook`exemple suivant supprime un webhook du projet spécifié CodeBuild .

```
aws codebuild delete-webhook --project-name my-project
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Arrêter l'exécution automatique des builds (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteWebhookà la section Référence des AWS CLI commandes.

`describe-code-coverages`

L'exemple de code suivant montre comment utiliser`describe-code-coverages`.

AWS CLI

Pour obtenir des informations détaillées sur la couverture du code, testez AWS CodeBuild.

L'`describe-code-coverages`exemple suivant permet d'obtenir des informations sur les résultats du test de couverture du code dans le rapport spécifié.

```
aws codebuild describe-code-coverages \
    --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Sortie :

```
{
    "codeCoverages": [
        {
            "id": "20a0adcc-db13-4b66-804b-ecaf9f852855",
            "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-group-name>:<report-ID>",
            "filePath": "<source-file-1-path>",
            "lineCoveragePercentage": 83.33,
            "linesCovered": 5,
```

```
            "linesMissed": 1,
            "branchCoveragePercentage": 50.0,
            "branchesCovered": 1,
            "branchesMissed": 1,
            "expired": "2020-11-20T21:22:45+00:00"
        },
        {
            "id": "0887162d-bf57-4cf1-a164-e432373d1a83",
            "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-
group-name>:<report-ID>",
            "filePath": "<source-file-2-path>",
            "lineCoveragePercentage": 90.9,
            "linesCovered": 10,
            "linesMissed": 1,
            "branchCoveragePercentage": 50.0,
            "branchesCovered": 1,
            "branchesMissed": 1,
            "expired": "2020-11-20T21:22:45+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez les rapports sur la couverture du code dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeCodeCoveragesà la section Référence des AWS CLI commandes.

`describe-test-cases`

L'exemple de code suivant montre comment utiliser`describe-test-cases`.

AWS CLI

Pour obtenir des informations détaillées sur les cas de test dans AWS CodeBuild.

L'`describe-test-cases`exemple suivant permet d'obtenir des informations sur les cas de test dans le rapport spécifié.

```
aws codebuild describe-test-cases \
    --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-
name>:<report-ID>
```

Sortie :

```
{
    "testCases": [
        {
            "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-
group-name>:<report-ID>",
            "testRawDataPath": "<test-report-path>",
            "prefix": "NUnit.Tests.Assemblies.MockTestFixture",
            "name": "NUnit.Tests.Assemblies.MockTestFixture.NotRunnableTest",
            "status": "ERROR",
            "durationInNanoSeconds": 0,
            "message": "No arguments were provided\n",
            "expired": "2020-11-20T17:52:10+00:00"
        },
        {
            "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-
group-name>:<report-ID>",
            "testRawDataPath": "<test-report-path>",
            "prefix": "NUnit.Tests.Assemblies.MockTestFixture",
            "name": "NUnit.Tests.Assemblies.MockTestFixture.TestWithException",
            "status": "ERROR",
            "durationInNanoSeconds": 0,
            "message": "System.ApplicationException : Intentional Exception
\nat NUnit.Tests.Assemblies.MockTestFixture.MethodThrowsException()\nat
 NUnit.Tests.Assemblies.MockTestFixture.TestWithException()\n\n",
            "expired": "2020-11-20T17:52:10+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des rapports de test AWS CodeBuild dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeTestCasesà la section Référence des AWS CLI commandes.

`import-source-credentials`

L'exemple de code suivant montre comment utiliser`import-source-credentials`.

AWS CLI

Connectez un AWS CodeBuild utilisateur à un fournisseur source en important les informations d'identification du fournisseur source.

L'`import-source-credentials`exemple suivant importe un jeton pour un dépôt Bitbucket qui utilise BASIC_AUTH comme type d'authentification.

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-
type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

Sortie :

```
{
    "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"
}
```

Pour plus d'informations, voir Connect Source Providers with Access Tokens (CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ImportSourceCredentialsà la section Référence des AWS CLI commandes.

`invalidate-project-cache`

L'exemple de code suivant montre comment utiliser`invalidate-project-cache`.

AWS CLI

Pour réinitialiser le cache d'un projet de AWS CodeBuild génération.

L'`invalidate-project-cache`exemple suivant réinitialise le cache pour le CodeBuild projet spécifié.

```
aws codebuild invalidate-project-cache --project-name my-project
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Build Caching in CodeBuild dans le guide de l'AWS CodeBuild utilisateur.

- Pour plus de détails sur l'API, reportez-vous InvalidateProjectCacheà la section Référence des AWS CLI commandes.

`list-build-batches-for-project`

L'exemple de code suivant montre comment utiliser`list-build-batches-for-project`.

AWS CLI

Pour répertorier les versions par lots pour un projet de construction spécifique dans AWS CodeBuild.

L'`list-build-batches-for-project`exemple suivant répertorie les versions CodeBuild par lots pour le projet spécifié.

```
aws codebuild list-build-batches-for-project \
    --project-name "<project-name>"
```

Sortie :

```
{
    "ids": [
        "<project-name>:<batch-ID>",
        "<project-name>:<batch-ID>"
    ]
}
```

Pour plus d'informations, consultez la section Batch builds in AWS CodeBuild dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListBuildBatchesForProjectà la section Référence des AWS CLI commandes.

`list-build-batches`

L'exemple de code suivant montre comment utiliser`list-build-batches`.

AWS CLI

Pour répertorier les intégrations par lots AWS CodeBuild.

L'`list-build-batches`exemple suivant répertorie les builds CodeBuild par lots pour le compte courant.

```
aws codebuild list-build-batches
```

Sortie :

```
{
    "ids": [
        "<project-name>:<batch-ID>",
        "<project-name>:<batch-ID>"
    ]
}
```

Pour plus d'informations, consultez Batch builds in AWS CodeBuild < https://docs.aws.amazon.com/codebuild/ latest/userguide/batch -build.html>) __ dans le guide de l'utilisateur.AWS CodeBuild

- Pour plus de détails sur l'API, reportez-vous ListBuildBatchesà la section Référence des AWS CLI commandes.

`list-builds-for-project`

L'exemple de code suivant montre comment utiliser`list-builds-for-project`.

AWS CLI

Pour afficher la liste des versions d'un projet de AWS CodeBuild construction.

L'`list-builds-for-project`exemple suivant répertorie la construction IDs par ordre décroissant pour le projet de CodeBuild construction spécifié.

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-
order DESCENDING
```

Sortie :

```
{
    "ids": [
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-11111example",
```

```
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-22222example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-33333example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-44444example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-55555example"
    ]
}
```

Pour plus d'informations, voir Afficher la liste des versions IDs pour un projet de construction (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur

- Pour plus de détails sur l'API, reportez-vous ListBuildsForProjectà la section Référence des AWS CLI commandes.

`list-builds`

L'exemple de code suivant montre comment utiliser`list-builds`.

AWS CLI

Pour obtenir une liste des AWS CodeBuild versions IDs.

L'`list-builds`exemple suivant obtient une liste de CodeBuild IDs triés par ordre croissant.

```
aws codebuild list-builds --sort-order ASCENDING
```

La sortie inclut une `nextToken` valeur qui indique qu'il y a plus de sorties disponibles.

```
{
    "nextToken": "4AEA6u7J...The full token has been omitted for
 brevity...MzY2OA==",
    "ids": [
        "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
        "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
            ... The full list of build IDs has been omitted for brevity ...
        "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
    ]
}
```

Réexécutez cette commande et fournissez la `nextToken` valeur de la réponse précédente en tant que paramètre pour obtenir la partie suivante de la sortie. Répétez l'opération jusqu'à ce que vous ne receviez `nextToken` aucune valeur dans la réponse.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Partie suivante de la sortie :

```
{
    "ids": [
        "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
        "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
            ... The full list of build IDs has been omitted for brevity ...
        "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
    ]
}
```

Pour plus d'informations, voir Afficher une liste de versions IDs (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur

- Pour plus de détails sur l'API, reportez-vous ListBuildsà la section Référence des AWS CLI commandes.

`list-curated-environment-images`

L'exemple de code suivant montre comment utiliser`list-curated-environment-images`.

AWS CLI

Pour obtenir une liste des images Docker gérées par AWS CodeBuild celles-ci, vous pouvez les utiliser pour vos builds.

L'`list-curated-environment-images`exemple suivant répertorie les images Docker gérées par CodeBuild qui peuvent être utilisées pour les builds. :

```
aws codebuild list-curated-environment-images
```

Sortie :

```
{
    "platforms": [
        {
            "platform": "AMAZON_LINUX",
            "languages": [
```

```
                {
                    "language": "JAVA",
                    "images": [
                        {
                            "description": "AWS ElasticBeanstalk - Java 7 Running on
 Amazon Linux 64bit v2.1.3",
                            "name": "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3",
                            "versions": [
                                "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3-1.0.0"
                            ]
                        },
                        {
                            "description": "AWS ElasticBeanstalk - Java 8 Running on
 Amazon Linux 64bit v2.1.3",
                            "name": "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3",
                            "versions": [
                                "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3-1.0.0"
                            ]
                        },
                        ... LIST TRUNCATED FOR BREVITY ...
                    ]
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Images Docker fournies par CodeBuild dans le guide de l'AWS CodeBuild utilisateur

- Pour plus de détails sur l'API, reportez-vous ListCuratedEnvironmentImagesà la section Référence des AWS CLI commandes.

`list-projects`

L'exemple de code suivant montre comment utiliser`list-projects`.

AWS CLI

Pour obtenir une liste des noms de projets de AWS CodeBuild construction.

L'`list-projects`exemple suivant obtient une liste de projets de CodeBuild construction triés par nom dans l'ordre croissant.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

La sortie inclut une `nextToken` valeur qui indique qu'il y a plus de sorties disponibles.

```
{
    "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
    "projects": [
        "codebuild-demo-project",
        "codebuild-demo-project2",
            ... The full list of build project names has been omitted for
 brevity ...
        "codebuild-demo-project99"
    ]
}
```

Réexécutez cette commande et fournissez la `nextToken` valeur de la réponse précédente en tant que paramètre pour obtenir la partie suivante de la sortie. Répétez l'opération jusqu'à ce que vous ne receviez `nextToken` aucune valeur dans la réponse.

```
aws codebuild list-projects  --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
    "projects": [
        "codebuild-demo-project100",
        "codebuild-demo-project101",

  ... The full list of build project names has been omitted for brevity ...
        "codebuild-demo-project122"

  ]
}
```

Pour plus d'informations, voir Afficher une liste de noms de projets de construction (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListProjectsà la section Référence des AWS CLI commandes.

`list-report-groups`

L'exemple de code suivant montre comment utiliser`list-report-groups`.

AWS CLI

Pour obtenir la liste du groupe ARNs de rapports AWS CodeBuild.

L'`list-report-groups`exemple suivant permet de récupérer le groupe de rapports ARNs pour le compte de la région.

```
aws codebuild list-report-groups
```

Sortie :

```
{
    "reportGroups": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListReportGroupsà la section Référence des AWS CLI commandes.

`list-reports-for-report-group`

L'exemple de code suivant montre comment utiliser`list-reports-for-report-group`.

AWS CLI

Pour obtenir la liste des rapports d'un groupe de rapports dans AWS CodeBuild.

L'`list-report-for-report-groups`exemple suivant extrait les rapports du groupe de rapports spécifié pour le compte de la région.

```
aws codebuild list-reports-for-report-group \
```

```
    --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

Sortie :

```
{
    "reports": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-1",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-2",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-3"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListReportsForReportGroupà la section Référence des AWS CLI commandes.

`list-reports`

L'exemple de code suivant montre comment utiliser`list-reports`.

AWS CLI

Pour obtenir une liste des rapports relatifs au compte courant dans AWS CodeBuild.

L'`list-reports`exemple suivant récupère les rapports ARNs du compte courant.

```
aws codebuild list-reports
```

Sortie :

```
{
    "reports": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report
 ID>",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report
 ID>",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report
 ID>"
```

```
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListReportsà la section Référence des AWS CLI commandes.

`list-shared-projects`

L'exemple de code suivant montre comment utiliser`list-shared-projects`.

AWS CLI

Pour répertorier le projet partagé dans AWS CodeBuild.

L'`list-shared-projects`exemple suivant répertorie les projets CodeBuild partagés disponibles pour le compte courant.

```
aws codebuild list-shared-projects
```

Sortie :

```
{
    "projects": [
        "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-
name-1>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-name-2>"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de projets partagés dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListSharedProjectsà la section Référence des AWS CLI commandes.

`list-shared-report-groups`

L'exemple de code suivant montre comment utiliser`list-shared-report-groups`.

AWS CLI

Pour obtenir la liste du groupe ARNs de rapports partagé AWS CodeBuild.

L'`list-shared-report-groups`exemple suivant permet de récupérer le groupe de rapports ARNs pour le compte de la région.

```
aws codebuild list-shared-report-groups
```

Sortie :

```
{
    "reportGroups": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListSharedReportGroupsà la section Référence des AWS CLI commandes.

`list-source-credentials`

L'exemple de code suivant montre comment utiliser`list-source-credentials`.

AWS CLI

Pour consulter la liste des sourceCredentialsObjects

L'`list-source-credentials`exemple suivant répertorie les jetons pour un AWS compte connecté à un compte Bitbucket et à un GitHub compte. Chaque `sourceCredentialsInfos` objet de la réponse contient les informations d'identification de la source connectée.

```
aws codebuild list-source-credentials
```

Sortie :

```
{
```

```
    "sourceCredentialsInfos": [
        {
            "serverType": "BITBUCKET",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket",
            "authType": "BASIC_AUTH"
        },
        {
            "serverType": "GITHUB",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:token/github",
            "authType": "OAUTH"
        }
    ]
}
```

Pour plus d'informations, voir Connect Source Providers with Access Tokens (CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListSourceCredentialsà la section Référence des AWS CLI commandes.

`retry-build-batch`

L'exemple de code suivant montre comment utiliser`retry-build-batch`.

AWS CLI

Pour réessayer une intégration par lots qui a échoué. AWS CodeBuild

L'`retry-build-batch`exemple suivant redémarre la génération par lots spécifiée.

```
aws codebuild retry-build-batch \
    --id <project-name>:<batch-ID>
```

Sortie :

```
{
    "buildBatch": {
        "id": "<project-name>:<batch-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-
name>:<batch-ID>",
        "startTime": "2020-10-21T17:26:23.099000+00:00",
        "currentPhase": "SUBMITTED",
        "buildBatchStatus": "IN_PROGRESS",
```

```
        "resolvedSourceVersion": "3a9e11cb419e8fff14b03883dc4e64f6155aaa7e",
        "projectName": "<project-name>",
        "phases": [
            {
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T17:26:23.099000+00:00",
                "endTime": "2020-10-21T17:26:23.457000+00:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "DOWNLOAD_BATCHSPEC",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T17:26:23.457000+00:00",
                "endTime": "2020-10-21T17:26:54.902000+00:00",
                "durationInSeconds": 31
            },
            {
                "phaseType": "IN_PROGRESS",
                "phaseStatus": "CLIENT_ERROR",
                "startTime": "2020-10-21T17:26:54.902000+00:00",
                "endTime": "2020-10-21T17:28:16.060000+00:00",
                "durationInSeconds": 81
            },
            {
                "phaseType": "FAILED",
                "phaseStatus": "RETRY",
                "startTime": "2020-10-21T17:28:16.060000+00:00",
                "endTime": "2020-10-21T17:29:39.709000+00:00",
                "durationInSeconds": 83
            },
            {
                "phaseType": "SUBMITTED",
                "startTime": "2020-10-21T17:29:39.709000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "https://github.com/strohm-a/<project-name>-graph.git",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
```

```
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "complete": false,
        "initiator": "<username>",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
        "buildBatchNumber": 4,
        "buildBatchConfig": {
            "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
            "restrictions": {
                "maximumBuildsAllowed": 100
            },
            "timeoutInMins": 480
        },
        "buildGroups": [
            {
```

```
                "identifier": "DOWNLOAD_SOURCE",
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:23.889000+00:00",
                    "buildStatus": "SUCCEEDED",
                    "primaryArtifact": {
                        "type": "no_artifacts",
                        "identifier": "DOWNLOAD_SOURCE"
                    },
                    "secondaryArtifacts": []
                }
            },
            {
                "identifier": "linux_small",
                "dependsOn": [],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:55.115000+00:00",
                    "buildStatus": "FAILED",
                    "primaryArtifact": {
                        "type": "no_artifacts",
                        "identifier": "linux_small"
                    },
                    "secondaryArtifacts": []
                }
            },
            {
                "identifier": "linux_medium",
                "dependsOn": [
                    "linux_small"
                ],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:54.594000+00:00",
                    "buildStatus": "STOPPED"
                }
            },
            {
```

```
                "identifier": "linux_large",
                "dependsOn": [
                    "linux_medium"
                ],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:54.701000+00:00",
                    "buildStatus": "STOPPED"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Batch builds in AWS CodeBuild dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous RetryBuildBatchà la section Référence des AWS CLI commandes.

`retry-build`

L'exemple de code suivant montre comment utiliser`retry-build`.

AWS CLI

Pour réessayer une intégration qui a échoué. AWS CodeBuild

L'`retry-build`exemple suivant redémarre le build spécifié.

```
aws codebuild retry-build \
    --id <project-name>:<build-ID>
```

Sortie :

```
{
    "build": {
        "id": "<project-name>:<build-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-
name>:<build-ID>",
        "buildNumber": 9,
```

```
        "startTime": "2020-10-21T17:51:38.161000+00:00",
        "currentPhase": "QUEUED",
        "buildStatus": "IN_PROGRESS",
        "projectName": "<project-name>",
        "phases": [
            {
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T17:51:38.161000+00:00",
                "endTime": "2020-10-21T17:51:38.210000+00:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "QUEUED",
                "startTime": "2020-10-21T17:51:38.210000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "<GitHub-repo-URL>",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
```

```
        "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-role-
name>",
        "logs": {
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=<region-ID>#logEvent:group=null;stream=null",
            "cloudWatchLogsArn": "arn:aws:logs:<region-ID>:<account-ID>:log-
group:null:log-stream:null",
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "timeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "buildComplete": false,
        "initiator": "<username>",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3"
    }
}
```

Pour plus d'informations, consultez la section Batch builds in AWS CodeBuild dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous RetryBuildà la section Référence des AWS CLI commandes.

`start-build-batch`

L'exemple de code suivant montre comment utiliser`start-build-batch`.

AWS CLI

Pour démarrer une intégration par lots AWS CodeBuild.

L'`start-build-batch`exemple suivant démarre une génération par lots du projet spécifié.

```
aws codebuild start-build-batch \
    --project-name <project-name>
```

Sortie :

```
{
    "buildBatch": {
        "id": "<project-name>:<batch-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-
name>:<batch-ID>",
        "startTime": "2020-10-21T16:54:24.740000+00:00",
        "currentPhase": "SUBMITTED",
        "buildBatchStatus": "IN_PROGRESS",
        "projectName": "<project-name>",
        "source": {
            "type": "GITHUB",
            "location": "<GitHub-repo-URL>",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
```

```
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "complete": false,
        "initiator": "<username>",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
        "buildBatchNumber": 3,
        "buildBatchConfig": {
            "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-
role-name>",
            "restrictions": {
                "maximumBuildsAllowed": 100
            },
            "timeoutInMins": 480
        }
    }
}
```

Pour plus d'informations, consultez la section Batch builds in AWS CodeBuild dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StartBuildBatchà la section Référence des AWS CLI commandes.

`start-build`

L'exemple de code suivant montre comment utiliser`start-build`.

AWS CLI

Pour commencer à exécuter la compilation d'un projet de AWS CodeBuild construction.

L'`start-build`exemple suivant lance une compilation pour le CodeBuild projet spécifié. La génération remplace à la fois le paramètre du projet concernant le nombre de minutes pendant lesquelles le build peut être mis en file d'attente avant son expiration et les paramètres des artefacts du projet.

```
aws codebuild start-build \
    --project-name "my-demo-project" \
    --queued-timeout-in-minutes-override 5 \
```

```
    --artifacts-override {"\"type\": \"S3\",\"location\": \"arn:aws:s3:::artifacts-
override\",\"overrideArtifactName\":true"}
```

Sortie :

```
{
    "build": {
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "buildStatus": "IN_PROGRESS",
        "buildComplete": false,
        "projectName": "my-demo-project",
        "timeoutInMinutes": 60,
        "source": {
            "insecureSsl": false,
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip"
        },
        "queuedTimeoutInMinutes": 5,
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "currentPhase": "QUEUED",
        "startTime": 1556905683.568,
        "environment": {
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "environmentVariables": [],
            "type": "LINUX_CONTAINER",
            "privilegedMode": false,
            "image": "aws/codebuild/standard:1.0",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "phases": [
            {
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556905683.568,
                "phaseType": "SUBMITTED",
                "durationInSeconds": 0,
                "endTime": 1556905684.524
            },
            {
                "startTime": 1556905684.524,
                "phaseType": "QUEUED"
            }
```

```
        ],
        "logs": {
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=null;stream=null"
        },
        "artifacts": {
            "encryptionDisabled": false,
            "location": "arn:aws:s3:::artifacts-override/my-demo-project",
            "overrideArtifactName": true
        },
        "cache": {
            "type": "NO_CACHE"
        },
        "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
        "initiator": "my-aws-account-name",
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
    }
}
```

Pour plus d'informations, voir Run a Build (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StartBuildà la section Référence des AWS CLI commandes.

`stop-build-batch`

L'exemple de code suivant montre comment utiliser`stop-build-batch`.

AWS CLI

Pour arrêter l'intégration d'un lot en cours. AWS CodeBuild

L'`stop-build-batch`exemple suivant arrête la génération par lots spécifiée.

```
aws codebuild stop-build-batch \
    --id <project-name>:<batch-ID>
```

Sortie :

```
{
    "buildBatch": {
```

```
        "id": "<project-name>:<batch-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-
name>:<batch-ID>",
        "startTime": "2020-10-21T16:54:24.740000+00:00",
        "endTime": "2020-10-21T16:56:05.152000+00:00",
        "currentPhase": "STOPPED",
        "buildBatchStatus": "STOPPED",
        "resolvedSourceVersion": "aef7744ed069c51098e15c360f4102cd2cd1ad64",
        "projectName": "<project-name>",
        "phases": [
            {
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T16:54:24.740000+00:00",
                "endTime": "2020-10-21T16:54:25.039000+00:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "DOWNLOAD_BATCHSPEC",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T16:54:25.039000+00:00",
                "endTime": "2020-10-21T16:54:56.583000+00:00",
                "durationInSeconds": 31
            },
            {
                "phaseType": "IN_PROGRESS",
                "phaseStatus": "STOPPED",
                "startTime": "2020-10-21T16:54:56.583000+00:00",
                "endTime": "2020-10-21T16:56:05.152000+00:00",
                "durationInSeconds": 68
            },
            {
                "phaseType": "STOPPED",
                "startTime": "2020-10-21T16:56:05.152000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "<GitHub-repo-URL>",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
```

```
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "complete": true,
        "initiator": "Strohm",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
        "buildBatchNumber": 3,
        "buildBatchConfig": {
            "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
            "restrictions": {
                "maximumBuildsAllowed": 100
            },
            "timeoutInMins": 480
        },
        "buildGroups": [
            {
```

```
                "identifier": "DOWNLOAD_SOURCE",
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:25.468000+00:00",
                    "buildStatus": "SUCCEEDED",
                    "primaryArtifact": {
                        "type": "no_artifacts",
                        "identifier": "DOWNLOAD_SOURCE"
                    },
                    "secondaryArtifacts": []
                }
            },
            {
                "identifier": "linux_small",
                "dependsOn": [],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:56.833000+00:00",
                    "buildStatus": "IN_PROGRESS"
                }
            },
            {
                "identifier": "linux_medium",
                "dependsOn": [
                    "linux_small"
                ],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:56.211000+00:00",
                    "buildStatus": "PENDING"
                }
            },
            {
                "identifier": "linux_large",
                "dependsOn": [
                    "linux_medium"
                ],
                "ignoreFailure": false,
```

```
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:56.330000+00:00",
                    "buildStatus": "PENDING"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Batch builds in AWS CodeBuild dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StopBuildBatchà la section Référence des AWS CLI commandes.

`stop-build`

L'exemple de code suivant montre comment utiliser`stop-build`.

AWS CLI

Pour arrêter la construction d'un projet de AWS CodeBuild construction.

L'`stop-build`exemple suivant arrête la CodeBuild construction spécifiée.

```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

Sortie :

```
{
    "build": {
        "startTime": 1556906956.318,
        "initiator": "my-aws-account-name",
        "projectName": "my-demo-project",
        "currentPhase": "COMPLETED",
        "cache": {
            "type": "NO_CACHE"
        },
        "source": {
            "insecureSsl": false,
```

```
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
            "type": "S3"
        },
        "id": "my-demo-project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
        "endTime": 1556906974.781,
        "phases": [
            {
                "durationInSeconds": 0,
                "phaseType": "SUBMITTED",
                "endTime": 1556906956.935,
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556906956.318
            },
            {
                "durationInSeconds": 1,
                "phaseType": "QUEUED",
                "endTime": 1556906958.272,
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556906956.935
            },
            {
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED",
                "durationInSeconds": 14,
                "contexts": [
                    {
                        "message": "",
                        "statusCode": ""
                    }
                ],
                "endTime": 1556906972.847,
                "startTime": 1556906958.272
            },
            {
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED",
                "durationInSeconds": 0,
                "contexts": [
                        {
                        "message": "",
                        "statusCode": ""
                    }
                ],
```

```
                "endTime": 1556906973.552,
                "startTime": 1556906972.847
            },
            {
                "phaseType": "INSTALL",
                "phaseStatus": "SUCCEEDED",
                "durationInSeconds": 0,
                "contexts": [
                    {
                        "message": "",
                        "statusCode": ""
                    }
                ],
                "endTime": 1556906973.75,
                "startTime": 1556906973.552
            },
            {
                "phaseType": "PRE_BUILD",
                "phaseStatus": "SUCCEEDED",
                "durationInSeconds": 0,
                "contexts": [
                    {
                        "message": "",
                        "statusCode": ""
                    }
                ],
                "endTime": 1556906973.937,
                "startTime": 1556906973.75
            },
            {
                "durationInSeconds": 0,
                "phaseType": "BUILD",
                "endTime": 1556906974.781,
                "phaseStatus": "STOPPED",
                "startTime": 1556906973.937
            },
            {
                "phaseType": "COMPLETED",
                "startTime": 1556906974.781
            }
        ],
        "artifacts": {
            "location": "arn:aws:s3:::artifacts-override/my-demo-project",
            "encryptionDisabled": false,
```

```
            "overrideArtifactName": true
        },
        "buildComplete": true,
        "buildStatus": "STOPPED",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "queuedTimeoutInMinutes": 5,
        "timeoutInMinutes": 60,
        "environment": {
            "type": "LINUX_CONTAINER",
            "environmentVariables": [],
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "privilegedMode": false,
            "image": "aws/codebuild/standard:1.0",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logs": {
            "streamName": "1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-demo-project;stream=1a2b3c4d-5678-90ab-
cdef-11111EXAMPLE",
            "groupName": "/aws/codebuild/my-demo-project"
        },
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Stop a Build (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StopBuildà la section Référence des AWS CLI commandes.

`update-project`

L'exemple de code suivant montre comment utiliser`update-project`.

AWS CLI

Pour modifier les paramètres d'un projet de AWS CodeBuild construction.

L'`update-project`exemple suivant modifie les paramètres du projet de CodeBuild construction spécifié nommé my-demo-project.

```
aws codebuild update-project --name "my-demo-project" \
    --description "This project is updated" \
    --source "{\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-input-bucket/my-source-2.zip\"}" \
    --artifacts {"\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-output-bucket-2\""} \
    --environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/standard:1.0\",\"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
    --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role"
```

La sortie affiche les paramètres mis à jour.

```
{
    "project": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
        "environment": {
            "privilegedMode": false,
            "environmentVariables": [],
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "queuedTimeoutInMinutes": 480,
        "description": "This project is updated",
        "artifacts": {
            "packaging": "NONE",
            "name": "my-demo-project",
            "type": "S3",
            "namespaceType": "NONE",
            "encryptionDisabled": false,
            "location": "codebuild-us-west-2-123456789012-output-bucket-2"
        },
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "badge": {
            "badgeEnabled": false
        },
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",
```

```
        "lastModified": 1556840545.967,
        "tags": [],
        "timeoutInMinutes": 60,
        "created": 1556839783.274,
        "name": "my-demo-project",
        "cache": {
            "type": "NO_CACHE"
        },
        "source": {
            "type": "S3",
            "insecureSsl": false,
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source-2.zip"
        }
    }
}
```

Pour plus d'informations, voir Modifier les paramètres d'un projet de construction (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur

- Pour plus de détails sur l'API, reportez-vous UpdateProjectà la section Référence des AWS CLI commandes.

`update-report-group`

L'exemple de code suivant montre comment utiliser`update-report-group`.

AWS CLI

Pour mettre à jour un groupe de rapports dans AWS CodeBuild.

L'`update-report-group`exemple suivant change le type d'exportation du groupe de rapports en « NO_EXPORT ».

```
aws codebuild update-report-group \
    --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-
group \
    --export-config="exportConfigType=NO_EXPORT"
```

Sortie :

```
{
```

```
    "reportGroup": {
        "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-
report-group",
        "name": "cli-created-report-group",
        "type": "TEST",
        "exportConfig": {
            "exportConfigType": "NO_EXPORT"
        },
        "created": 1602020686.009,
        "lastModified": 1602021033.454,
        "tags": []
    }
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de rapports dans le Guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateReportGroupà la section Référence des AWS CLI commandes.

`update-webhook`

L'exemple de code suivant montre comment utiliser`update-webhook`.

AWS CLI

Pour mettre à jour le webhook d'un projet AWS CodeBuild

L'`update-webhook`exemple suivant met à jour un webhook pour le CodeBuild projet spécifié avec deux groupes de filtres. Le `--rotate-secret` paramètre indique qu'il GitHub faut faire pivoter la clé secrète du projet chaque fois qu'une modification de code déclenche une génération. Le premier groupe de filtres spécifie les demandes d'extraction créées, mises à jour ou rouvertes sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/master$` et des références principales qui correspondent à `^refs/heads/myBranch$`. Le deuxième groupe de filtres spécifie les requêtes push sur les branches dont les noms de référence Git ne correspondent pas à l'expression régulière`^refs/heads/myBranch$`.

```
aws codebuild update-webhook \
    --project-name Project2 \
    --rotate-secret \
    --filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"PULL_REQUEST_CREATED,
 PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED\"},{\"type\":\"HEAD_REF\",\"pattern
```

```
\":\"^refs/heads/myBranch$\",\"excludeMatchedPattern\":true},{\"type\":\"BASE_REF
\",\"pattern\":\"^refs/heads/master$\",\"excludeMatchedPattern\":true}],[{\"type\":
\"EVENT\",\"pattern\":\"PUSH\"},{\"type\":\"HEAD_REF\",\"pattern\":\"^refs/heads/
myBranch$\",\"excludeMatchedPattern\":true}]]"
```

Sortie :

```
{
    "webhook": {
        "filterGroups": [
            [
                {
                    "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED",
                    "type": "EVENT"
                },
                {
                    "excludeMatchedPattern": true,
                    "pattern": "refs/heads/myBranch$",
                    "type": "HEAD_REF"
                },
                {
                    "excludeMatchedPattern": true,
                    "pattern": "refs/heads/master$",
                    "type": "BASE_REF"
                }
            ],
            [
                {
                    "pattern": "PUSH",
                    "type": "EVENT"
                },
                {
                    "excludeMatchedPattern": true,
                    "pattern": "refs/heads/myBranch$",
                    "type": "HEAD_REF"
                }
            ]
        ],
        "lastModifiedSecret": 1556312220.133
    }
}
```

Pour plus d'informations, voir Modifier les paramètres d'un projet de construction (AWS CLI) dans le guide de AWS CodeBuild l'utilisateur

- Pour plus de détails sur l'API, reportez-vous UpdateWebhookà la section Référence des AWS CLI commandes.

CodeCommit exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CodeCommit.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-approval-rule-template-with-repository`

L'exemple de code suivant montre comment utiliser`associate-approval-rule-template-with-repository`.

AWS CLI

Pour associer un modèle de règle d'approbation à un référentiel

L'`associate-approval-rule-template-with-repository`exemple suivant associe le modèle de règle d'approbation spécifié à un référentiel nomméMyDemoRepo.

```
aws codecommit associate-approval-rule-template-with-repository \
    --repository-name MyDemoRepo  \
    --approval-rule-template-name 2-approver-rule-for-main
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Associer un modèle de règle d'approbation à un référentiel dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous AssociateApprovalRuleTemplateWithRepositoryà la section Référence des AWS CLI commandes.

`batch-associate-approval-rule-template-with-repositories`

L'exemple de code suivant montre comment utiliser`batch-associate-approval-rule-template-with-repositories`.

AWS CLI

Pour associer un modèle de règle d'approbation à plusieurs référentiels en une seule opération

L'`batch-associate-approval-rule-template-with-repositories`exemple suivant associe le modèle de règle d'approbation spécifié à des référentiels nommés `MyDemoRepo` et`MyOtherDemoRepo`.

Remarque : Les modèles de règles d'approbation sont spécifiques à la AWS région dans laquelle ils sont créés. Ils ne peuvent être associés qu'aux référentiels de cette AWS région.

```
aws codecommit batch-associate-approval-rule-template-with-repositories \
    --repository-names MyDemoRepo, MyOtherDemoRepo  \
    --approval-rule-template-name 2-approver-rule-for-main
```

Sortie :

```
{
    "associatedRepositoryNames": [
        "MyDemoRepo",
        "MyOtherDemoRepo"
    ],
    "errors": []
}
```

Pour plus d'informations, voir Associer un modèle de règle d'approbation à un référentiel dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchAssociateApprovalRuleTemplateWithRepositoriesà la section Référence des AWS CLI commandes.

batch-describe-merge-conflicts

L'exemple de code suivant montre comment utiliserbatch-describe-merge-conflicts.

AWS CLI

Pour obtenir des informations sur les conflits de fusion dans tous les fichiers ou dans un sous-ensemble de fichiers lors d'une fusion entre deux spécificateurs de validation

L'batch-describe-merge-conflictsexemple suivant détermine les conflits de fusion liés à la fusion d'une branche source nommée feature-randomizationfeature avec une branche de destination nommée main selon la THREE_WAY_MERGE stratégie dans un référentiel nomméMyDemoRepo.

```
aws codecommit batch-describe-merge-conflicts \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "conflicts": [
        {
            "conflictMetadata": {
                "filePath": "readme.md",
                "fileSizes": {
                    "source": 139,
                    "destination": 230,
                    "base": 85
                },
                "fileModes": {
                    "source": "NORMAL",
                    "destination": "NORMAL",
                    "base": "NORMAL"
                },
                "objectTypes": {
                    "source": "FILE",
                    "destination": "FILE",
                    "base": "FILE"
                },
```

```
                "numberOfConflicts": 1,
                "isBinaryFile": {
                    "source": false,
                    "destination": false,
                    "base": false
                },
                "contentConflict": true,
                "fileModeConflict": false,
                "objectTypeConflict": false,
                "mergeOperations": {
                    "source": "M",
                    "destination": "M"
                }
            },
            "mergeHunks": [
                {
                    "isConflict": true,
                    "source": {
                        "startLine": 0,
                        "endLine": 3,
                        "hunkContent": "VGhpcyBpEXAMPLE=="
                    },
                    "destination": {
                        "startLine": 0,
                        "endLine": 1,
                        "hunkContent": "VXNlIHRoEXAMPLE="
                    }
                }
            ]
        }
    ],
    "errors": [],
    "destinationCommitId": "86958e0aEXAMPLE",
    "sourceCommitId": "6ccd57fdEXAMPLE",
    "baseCommitId": "767b6958EXAMPLE"
}
```

Pour plus d'informations, consultez la section Résoudre les conflits liés à une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchDescribeMergeConflictsà la section Référence des AWS CLI commandes.

batch-disassociate-approval-rule-template-from-repositories

L'exemple de code suivant montre comment utiliserbatch-disassociate-approval-rule-template-from-repositories.

AWS CLI

Pour dissocier un modèle de règle d'approbation de plusieurs référentiels en une seule opération

L'batch-disassociate-approval-rule-template-from-repositoriesexemple suivant dissocie le modèle de règle d'approbation spécifié des référentiels nommés MyDemoRepo et. MyOtherDemoRepo

```
aws codecommit batch-disassociate-approval-rule-template-from-repositories \
    --repository-names MyDemoRepo, MyOtherDemoRepo  \
    --approval-rule-template-name 1-approval-rule-for-all pull requests
```

Sortie :

```
{
    "disassociatedRepositoryNames": [
        "MyDemoRepo",
        "MyOtherDemoRepo"
    ],
    "errors": []
}
```

Pour plus d'informations, voir Dissocier un modèle de règle d'approbation dans le guide de l'AWS CodeCommit utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchDisassociateApprovalRuleTemplateFromRepositoriesà la section Référence des AWS CLI commandes.

batch-get-commits

L'exemple de code suivant montre comment utiliserbatch-get-commits.

AWS CLI

Pour afficher les informations relatives à plusieurs validations

L'`batch-get-commits`exemple suivant affiche des détails sur les validations spécifiées.

```
aws codecommit batch-get-commits  \
    --repository-name MyDemoRepo  \
    --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

Sortie :

```
{
    "commits": [
      {
        "additionalData": "",
        "committer": {
            "date": "1508280564 -0800",
            "name": "Mary Major",
            "email": "mary_major@example.com"
        },
        "author": {
            "date": "1508280564 -0800",
            "name": "Mary Major",
            "email": "mary_major@example.com"
        },
        "commitId": "317f8570EXAMPLE",
        "treeId": "1f330709EXAMPLE",
        "parents": [
            "6e147360EXAMPLE"
        ],
        "message": "Change variable name and add new response element"
    },
    {
        "additionalData": "",
        "committer": {
            "date": "1508280542 -0800",
            "name": "Li Juan",
            "email": "li_juan@example.com"
        },
        "author": {
            "date": "1508280542 -0800",
            "name": "Li Juan",
            "email": "li_juan@example.com"
        },
        "commitId": "4c925148EXAMPLE",
        "treeId": "1f330709EXAMPLE",
```

```
        "parents": [
            "317f8570EXAMPLE"
        ],
        "message": "Added new class"
    }
}
```

Pour plus d'informations, voir Afficher les détails des validations dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous BatchGetCommitsà la section Référence des AWS CLI commandes.

`batch-get-repositories`

L'exemple de code suivant montre comment utiliser`batch-get-repositories`.

AWS CLI

Pour afficher les informations relatives à plusieurs référentiels

Cet exemple montre des détails sur plusieurs AWS CodeCommit référentiels.

```
aws codecommit batch-get-repositories \
    --repository-names MyDemoRepo MyOtherDemoRepo
```

Sortie :

```
{
    "repositoriesNotFound": [],
    "repositories": [
         {
            "creationDate": 1429203623.625,
            "defaultBranch": "main",
            "repositoryName": "MyDemoRepo",
            "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo",
            "lastModifiedDate": 1430783812.0869999,
            "repositoryDescription": "My demonstration repository",
            "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo",
```

```
            "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
            "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"
            "accountId": "111111111111"
        },
        {
            "creationDate": 1429203623.627,
            "defaultBranch": "main",
            "repositoryName": "MyOtherDemoRepo",
            "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyOtherDemoRepo",
            "lastModifiedDate": 1430783812.0889999,
            "repositoryDescription": "My other demonstration repository",
            "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/
MyOtherDemoRepo",
            "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE",
            "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"
            "accountId": "111111111111"
        }
    ],
    "repositoriesNotFound": []
}
```

- Pour plus de détails sur l'API, reportez-vous BatchGetRepositoriesà la section Référence des AWS CLI commandes.

`create-approval-rule-template`

L'exemple de code suivant montre comment utiliser`create-approval-rule-template`.

AWS CLI

Pour créer un modèle de règle d'approbation

L'`create-approval-rule-template`exemple suivant crée un modèle de règle d'approbation nommé `2-approver-rule-for-main ``. The template requires two users who assume the role of ``CodeCommitReview` pour approuver toute pull request avant qu'elle ne puisse être fusionnée avec la `main` branche.

```
aws codecommit create-approval-rule-template \
    --approval-rule-template-name 2-approver-rule-for-main \
    --approval-rule-template-description  "Requires two developers from the team to
 approve the pull request if the destination branch is main" \
```

```
    --approval-rule-template-content "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

Sortie :

```
{
    "approvalRuleTemplate": {
        "approvalRuleTemplateName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
        "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "approvalRuleTemplateDescription": "Requires two developers from the team to
 approve the pull request if the destination branch is main",
        "lastModifiedDate": 1571356106.936,
        "ruleContentSha256": "4711b576EXAMPLE"
    }
}
```

Pour plus d'informations, voir Création d'un modèle de règle d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateApprovalRuleTemplateà la section Référence des AWS CLI commandes.

`create-branch`

L'exemple de code suivant montre comment utiliser`create-branch`.

AWS CLI

Pour créer une branche

Cet exemple crée une branche dans un AWS CodeCommit référentiel. Cette commande génère une sortie uniquement en cas d'erreurs.

Commande :

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
 --commit-id 317f8570EXAMPLE
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous CreateBranchà la section Référence des AWS CLI commandes.

`create-commit`

L'exemple de code suivant montre comment utiliser`create-commit`.

AWS CLI

Pour créer un commit

L'`create-commit`exemple suivant montre comment créer un commit initial pour un dépôt qui ajoute un `readme.md` fichier à un dépôt nommé `MyDemoRepo` dans la `main` branche.

```
aws codecommit create-commit \
    --repository-name MyDemoRepo \
    --branch-name main \
    --put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

Sortie :

```
{
    "filesAdded": [
        {
            "blobId": "5e1c309d-EXAMPLE",
            "absolutePath": "readme.md",
            "fileMode": "NORMAL"
        }
    ],
    "commitId": "4df8b524-EXAMPLE",
    "treeId": "55b57003-EXAMPLE",
    "filesDeleted": [],
    "filesUpdated": []
}
```

Pour plus d'informations, consultez la section Créer un commit AWS CodeCommit dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateCommità la section Référence des AWS CLI commandes.

`create-pull-request-approval-rule`

L'exemple de code suivant montre comment utiliser`create-pull-request-approval-rule`.

AWS CLI

Pour créer une règle d'approbation pour une pull request

L'`create-pull-request-approval-rule`exemple suivant crée une règle d'approbation nommée `Require two approved approvers` pour la pull request spécifiée. La règle précise que deux approbations sont requises à partir d'un pool d'approbations. Le pool inclut tous les utilisateurs qui accèdent CodeCommit en assumant le rôle de `CodeCommitReview` dans le `123456789012` AWS compte. Il inclut également un utilisateur IAM ou un utilisateur fédéré nommé `Nikhil_Jayashankar` à partir du même AWS compte.

```
aws codecommit create-pull-request-approval-rule  \
    --approval-rule-name "Require two approved approvers"  \
    --approval-rule-content "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
 \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

Sortie :

```
{
    "approvalRule": {
        "approvalRuleName": "Require two approved approvers",
        "lastModifiedDate": 1570752871.932,
        "ruleContentSha256": "7c44e6ebEXAMPLE",
        "creationDate": 1570752871.932,
        "approvalRuleId": "aac33506-EXAMPLE",
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
 \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
```

```
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
    }
}
```

Pour plus d'informations, voir Création d'une règle d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreatePullRequestApprovalRuleà la section Référence des AWS CLI commandes.

`create-pull-request`

L'exemple de code suivant montre comment utiliser`create-pull-request`.

AWS CLI

Pour créer une pull request

L'`create-pull-request`exemple suivant crée une pull request nommée « Analyseur de difficulté de prononciation » avec la description « Veuillez revoir ces modifications d'ici mardi » qui cible la branche source « jane-branch » et doit être fusionnée avec la branche par défaut « main » dans un référentiel nommé « ». AWS CodeCommit MyDemoRepo

```
aws codecommit create-pull-request \
    --title "My Pull Request" \
    --description "Please review these changes by Tuesday" \
    --client-request-token 123Example \
    --targets repositoryName=MyDemoRepo,sourceReference=MyNewBranch
```

Sortie :

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
```

```
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd3d22fe-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::111111111111:user/Jane_Doe",
        "description": "Please review these changes by Tuesday",
        "title": "Pronunciation difficulty analyzer",
        "pullRequestTargets": [
            {
                "destinationCommit": "5d036259EXAMPLE",
                "destinationReference": "refs/heads/main",
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "317f8570EXAMPLE",
                "sourceReference": "refs/heads/jane-branch",
                "mergeMetadata": {
                    "isMerged": false
                }
            }
        ],
        "lastActivityDate": 1508962823.285,
        "pullRequestId": "42",
        "clientRequestToken": "123Example",
        "pullRequestStatus": "OPEN",
        "creationDate": 1508962823.285
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePullRequestà la section Référence des AWS CLI commandes.

`create-repository`

L'exemple de code suivant montre comment utiliser`create-repository`.

AWS CLI

Pour créer un référentiel

Cet exemple crée un référentiel et l'associe au AWS compte de l'utilisateur.

Commande :

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-
description "My demonstration repository"
```

Sortie :

```
{
    "repositoryMetadata": {
        "repositoryName": "MyDemoRepo",
                "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/
repos/MyDemoRepo",
                "lastModifiedDate": 1444766838.027,
        "repositoryDescription": "My demonstration repository",
                "cloneUrlHttp": "https://git-codecommit.us-east-1.amazonaws.com/v1/
repos/MyDemoRepo",
        "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
                "Arn": "arn:aws:codecommit:us-
east-1:111111111111EXAMPLE:MyDemoRepo",
        "accountId": "111111111111"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRepositoryà la section Référence des AWS CLI commandes.

`create-unreferenced-merge-commit`

L'exemple de code suivant montre comment utiliser`create-unreferenced-merge-commit`.

AWS CLI

Pour créer un commit non référencé qui représente le résultat de la fusion de deux spécificateurs de commit

L'`create-unreferenced-merge-commit`exemple suivant crée un commit qui représente les résultats d'une fusion entre une branche source nommée et une branche de destination nommée `bugfix-1234` `main` selon la stratégie THREE_WAY_MERGE dans un référentiel nommé. `MyDemoRepo`

```
aws codecommit create-unreferenced-merge-commit \
    --source-commit-specifier bugfix-1234 \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --repository-name MyDemoRepo \
    --name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "Testing the results of this merge."
```

Sortie :

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

Pour plus d'informations, consultez la section Résoudre les conflits liés à une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateUnreferencedMergeCommità la section Référence des AWS CLI commandes.

`credential-helper`

L'exemple de code suivant montre comment utiliser`credential-helper`.

AWS CLI

Pour configurer l'assistant d'identification inclus dans la CLI AWS avec AWS CodeCommit

L'`credential-helper`utilitaire n'est pas conçu pour être appelé directement depuis la AWS CLI. Il est plutôt destiné à être utilisé comme paramètre avec la `git config` commande de configuration de votre ordinateur local. Il permet à Git d'utiliser le protocole HTTPS et une version signée cryptographiquement de vos informations d'identification utilisateur IAM ou de votre rôle d' EC2 instance Amazon chaque fois que Git a besoin de s'authentifier AWS pour interagir avec des référentiels. CodeCommit

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

Sortie :

```
[credential]
    helper = !aws codecommit credential-helper $@
    UseHttpPath = true
```

Pour plus d'informations, consultez la section Configuration pour AWS CodeCommit l'utilisation d'autres méthodes dans le guide de AWS CodeCommit l'utilisateur. Lisez attentivement le contenu, puis suivez les procédures décrites dans l'une des rubriques suivantes : Pour les connexions HTTPS sous Linux, macOS ou Unix ou Pour les connexions HTTPS sous Windows dans le guide de l'AWS CodeCommit utilisateur.

- Pour plus de détails sur l'API, reportez-vous CredentialHelperà la section Référence des AWS CLI commandes.

`delete-approval-rule-template`

L'exemple de code suivant montre comment utiliser`delete-approval-rule-template`.

AWS CLI

Pour supprimer un modèle de règle d'approbation

L'`delete-approval-rule-template`exemple suivant supprime le modèle de règle d'approbation spécifié.

```
aws codecommit delete-approval-rule-template  \
    --approval-rule-template-name 1-approver-for-all-pull-requests
```

Sortie :

```
{
    "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

Pour plus d'informations, voir Supprimer un modèle de règle d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteApprovalRuleTemplateà la section Référence des AWS CLI commandes.

`delete-branch`

L'exemple de code suivant montre comment utiliser`delete-branch`.

AWS CLI

Pour supprimer une branche

Cet exemple montre comment supprimer une branche dans un AWS CodeCommit référentiel.

Commande :

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Sortie :

```
{
  "branch": {
      "commitId": "317f8570EXAMPLE",
      "branchName": "MyNewBranch"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBranchà la section Référence des AWS CLI commandes.

`delete-comment-content`

L'exemple de code suivant montre comment utiliser`delete-comment-content`.

AWS CLI

Pour supprimer le contenu d'un commentaire

Vous ne pouvez supprimer le contenu d'un commentaire que si vous l'avez créé. Cet exemple montre comment supprimer le contenu d'un commentaire dont l'ID généré par le système est de. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit delete-comment-content \
    --comment-id ff30b348EXAMPLEb9aa670f
```

Sortie :

```
{
    "comment": {
        "creationDate": 1508369768.142,
        "deleted": true,
        "lastModifiedDate": 1508369842.278,
        "clientRequestToken": "123Example",
        "commentId": "ff30b348EXAMPLEb9aa670f",
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "callerReactions": [],
        "reactionCounts":
        {
            "CLAP" : 1
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteCommentContentà la section Référence des AWS CLI commandes.

`delete-file`

L'exemple de code suivant montre comment utiliser`delete-file`.

AWS CLI

Pour supprimer un fichier

L'`delete-file`exemple suivant montre comment supprimer un fichier nommé à `README.md` partir d'une branche `main` dont le nom de validation est le plus récent `c5709475EXAMPLE` dans un référentiel nomméMyDemoRepo.

```
aws codecommit delete-file \
    --repository-name MyDemoRepo \
    --branch-name main \
    --file-path README.md \
    --parent-commit-id c5709475EXAMPLE
```

Sortie :

```
{
    "blobId":"559b44fEXAMPLE",
```

```
    "commitId":"353cf655EXAMPLE",
    "filePath":"README.md",
    "treeId":"6bc824cEXAMPLE"
}
```

Pour plus d'informations, voir Modifier ou supprimer un fichier AWS CodeCommit dans le guide de référence des AWS CodeCommit API.

- Pour plus de détails sur l'API, reportez-vous DeleteFileà la section Référence des AWS CLI commandes.

`delete-pull-request-approval-rule`

L'exemple de code suivant montre comment utiliser`delete-pull-request-approval-rule`.

AWS CLI

Pour supprimer une règle d'approbation pour une pull request

L'`delete-pull-request-approval-rule`exemple suivant supprime la règle d'approbation nommée `My Approval Rule` pour la pull request spécifiée.

```
aws codecommit delete-pull-request-approval-rule  \
    --approval-rule-name "My Approval Rule"  \
    --pull-request-id 15
```

Sortie :

```
{
    "approvalRuleId": "077d8e8a8-EXAMPLE"
}
```

Pour plus d'informations, voir Modifier ou supprimer une règle d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeletePullRequestApprovalRuleà la section Référence des AWS CLI commandes.

`delete-repository`

L'exemple de code suivant montre comment utiliser`delete-repository`.

AWS CLI

Pour supprimer un référentiel

Cet exemple montre comment supprimer un AWS CodeCommit dépôt.

Commande :

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

Sortie :

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteRepositoryà la section Référence des AWS CLI commandes.

`describe-merge-conflicts`

L'exemple de code suivant montre comment utiliser`describe-merge-conflicts`.

AWS CLI

Pour obtenir des informations détaillées sur les conflits de fusion

L'`describe-merge-conflicts`exemple suivant détermine les conflits de fusion pour un fichier nommé `readme.md` dans la branche source et la branche de destination spécifiées à l'aide de la stratégie THREE_WAY_MERGE.

```
aws codecommit describe-merge-conflicts \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --file-path readme.md \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "conflictMetadata": {
```

```
        "filePath": "readme.md",
        "fileSizes": {
            "source": 139,
            "destination": 230,
            "base": 85
        },
        "fileModes": {
            "source": "NORMAL",
            "destination": "NORMAL",
            "base": "NORMAL"
        },
        "objectTypes": {
            "source": "FILE",
            "destination": "FILE",
            "base": "FILE"
        },
        "numberOfConflicts": 1,
        "isBinaryFile": {
            "source": false,
            "destination": false,
            "base": false
        },
        "contentConflict": true,
        "fileModeConflict": false,
        "objectTypeConflict": false,
        "mergeOperations": {
            "source": "M",
            "destination": "M"
        }
    },
    "mergeHunks": [
        {
            "isConflict": true,
            "source": {
                "startLine": 0,
                "endLine": 3,
                "hunkContent": "VGhpcyBpEXAMPLE="
            },
            "destination": {
                "startLine": 0,
                "endLine": 1,
                "hunkContent": "VXNlIHRoEXAMPLE="
            }
        }
```

```
    ],
    "destinationCommitId": "86958e0aEXAMPLE",
    "sourceCommitId": "6ccd57fdEXAMPLE",
    "baseCommitId": "767b69580EXAMPLE"
}
```

Pour plus d'informations, consultez la section Résoudre les conflits liés à une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeMergeConflictsà la section Référence des AWS CLI commandes.

`describe-pull-request-events`

L'exemple de code suivant montre comment utiliser`describe-pull-request-events`.

AWS CLI

Pour afficher les événements dans une pull request

L'`describe-pull-request-events`exemple suivant récupère les événements d'une pull request avec l'ID « 8 ».

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Sortie :

```
{
    "pullRequestEvents": [
        {
            "pullRequestId": "8",
            "pullRequestEventType": "PULL_REQUEST_CREATED",
            "eventDate": 1510341779.53,
            "actor": "arn:aws:iam::111111111111:user/Zhang_Wei"
        },
        {
            "pullRequestStatusChangedEventMetadata": {
                "pullRequestStatus": "CLOSED"
            },
            "pullRequestId": "8",
            "pullRequestEventType": "PULL_REQUEST_STATUS_CHANGED",
            "eventDate": 1510341930.72,
```

```
            "actor": "arn:aws:iam::111111111111:user/Jane_Doe"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePullRequestEventsà la section Référence des AWS CLI commandes.

`disassociate-approval-rule-template-from-repository`

L'exemple de code suivant montre comment utiliser`disassociate-approval-rule-template-from-repository`.

AWS CLI

Pour dissocier un modèle de règle d'approbation d'un référentiel

L'`disassociate-approval-rule-template-from-repository`exemple suivant dissocie le modèle de règle d'approbation spécifié d'un référentiel nomméMyDemoRepo.

```
aws codecommit disassociate-approval-rule-template-from-repository \
    --repository-name MyDemoRepo  \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Dissocier un modèle de règle d'approbation dans le guide de l'AWS CodeCommit utilisateur.

- Pour plus de détails sur l'API, reportez-vous DisassociateApprovalRuleTemplateFromRepositoryà la section Référence des AWS CLI commandes.

`evaluate-pull-request-approval-rules`

L'exemple de code suivant montre comment utiliser`evaluate-pull-request-approval-rules`.

AWS CLI

Pour évaluer si toutes les règles d'approbation d'une pull request sont satisfaites

L'`evaluate-pull-request-approval-rules`exemple suivant évalue l'état des règles d'approbation de la pull request spécifiée. Dans cet exemple, une règle d'approbation n'ayant pas été respectée pour la pull request, la sortie de la commande affiche une `approved` valeur de`false`.

```
aws codecommit evaluate-pull-request-approval-rules \
    --pull-request-id 27  \
    --revision-id 9f29d167EXAMPLE
```

Sortie :

```
{
    "evaluation": {
        "approved": false,
        "approvalRulesNotSatisfied": [
            "Require two approved approvers"
        ],
        "overridden": false,
        "approvalRulesSatisfied": []
    }
}
```

Pour plus d'informations, consultez la section Fusionner une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous EvaluatePullRequestApprovalRulesà la section Référence des AWS CLI commandes.

`get-approval-rule-template`

L'exemple de code suivant montre comment utiliser`get-approval-rule-template`.

AWS CLI

Pour obtenir le contenu d'un modèle de règle d'approbation

L'`get-approval-rule-template`exemple suivant obtient le contenu d'un modèle de règle d'approbation nommé`1-approver-rule-for-all-pull-requests`.

```
aws codecommit get-approval-rule-template \
```

```
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Sortie :

```
{
    "approvalRuleTemplate": {
        "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "ruleContentSha256": "621181bbEXAMPLE",
        "lastModifiedDate": 1571356106.936,
        "creationDate": 1571356106.936,
        "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan",
        "approvalRuleTemplateId": "a29abb15-EXAMPLE",
        "approvalRuleTemplateDescription": "All pull requests must be approved by
 one developer on the team."
    }
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetApprovalRuleTemplateà la section Référence des AWS CLI commandes.

`get-blob`

L'exemple de code suivant montre comment utiliser`get-blob`.

AWS CLI

Pour afficher les informations relatives à un objet blob Git

L'`get-blob`exemple suivant extrait des informations sur un blob Git portant l'ID « 2eb4af3bExample » dans un référentiel nommé « ». AWS CodeCommit MyDemoRepo

```
aws codecommit get-blob  --repository-name MyDemoRepo  --blob-id 2eb4af3bEXAMPLE
```

Sortie :

```
{
```

```
    "content": "QSBCaW5hcnkgTGFyToEXAMPLE="
}
```

- Pour plus de détails sur l'API, reportez-vous GetBlobà la section Référence des AWS CLI commandes.

`get-branch`

L'exemple de code suivant montre comment utiliserget-branch.

AWS CLI

Pour obtenir des informations sur une succursale

Cet exemple permet d'obtenir des informations sur une branche d'un AWS CodeCommit référentiel.

Commande :

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Sortie :

```
{
  "BranchInfo": {
        "commitID": "317f8570EXAMPLE",
                "branchName": "MyNewBranch"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBranchà la section Référence des AWS CLI commandes.

`get-comment-reactions`

L'exemple de code suivant montre comment utiliserget-comment-reactions.

AWS CLI

Pour afficher les réactions des emoji à un commentaire

L'`get-comment-reactions`exemple suivant répertorie toutes les réactions emoji à un commentaire portant l'ID de`abcd1234EXAMPLEb5678efgh`. Si la police de votre shell prend en charge l'affichage de la version Emoji 1.0, la sortie de `emoji` l'emoji s'affiche.

```
aws codecommit get-comment-reactions \
    --comment-id abcd1234EXAMPLEb5678efgh
```

Sortie :

```
{
    "reactionsForComment": {
        [
           {
               "reaction": {
                    "emoji:"??",
                    "shortCode": "thumbsup",
                    "unicode": "U+1F44D"
                },
                "users": [
                    "arn:aws:iam::123456789012:user/Li_Juan",
                    "arn:aws:iam::123456789012:user/Mary_Major",
                    "arn:aws:iam::123456789012:user/Jorge_Souza"
                ]
            },
            {
                "reaction": {
                    "emoji": "??",
                    "shortCode": "thumbsdown",
                    "unicode": "U+1F44E"
                },
                "users": [
                    "arn:aws:iam::123456789012:user/Nikhil_Jayashankar"
                ]
            },
            {
                "reaction": {
                    "emoji": "??",
                    "shortCode": "confused",
                    "unicode": "U+1F615"
                },
                "users": [
                    "arn:aws:iam::123456789012:user/Saanvi_Sarkar"
                ]
```

```
            }
        ]
    }
}
```

Pour plus d'informations, voir Commenter un commit AWS CodeCommit dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetCommentReactionsà la section Référence des AWS CLI commandes.

`get-comment`

L'exemple de code suivant montre comment utiliser`get-comment`.

AWS CLI

Pour afficher les détails d'un commentaire

Cet exemple montre comment afficher les détails d'un commentaire dont l'ID de commentaire généré par le système est de. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit get-comment \
    --comment-id ff30b348EXAMPLEb9aa670f
```

Sortie :

```
{
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "123Example",
        "commentId": "ff30b348EXAMPLEb9aa670f",
        "content": "Whoops - I meant to add this comment to the line, but I don't
 see how to delete it.",
        "creationDate": 1508369768.142,
        "deleted": false,
        "commentId": "",
        "lastModifiedDate": 1508369842.278,
        "callerReactions": [],
        "reactionCounts":
        {
            "SMILE" : 6,
```

```
            "THUMBSUP" : 1
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetCommentà la section Référence des AWS CLI commandes.

`get-comments-for-compared-commit`

L'exemple de code suivant montre comment utiliser`get-comments-for-compared-commit`.

AWS CLI

Pour consulter les commentaires relatifs à un commit

Cet exemple montre comment afficher les commentaires relatifs à la comparaison entre deux validations dans un référentiel nomméMyDemoRepo.

```
aws codecommit get-comments-for-compared-commit \
    --repository-name MyDemoRepo \
    --before-commit-ID 6e147360EXAMPLE \
    --after-commit-id 317f8570EXAMPLE
```

Sortie :

```
{
    "commentsForComparedCommitData": [
        {
            "afterBlobId": "1f330709EXAMPLE",
            "afterCommitId": "317f8570EXAMPLE",
            "beforeBlobId": "80906a4cEXAMPLE",
            "beforeCommitId": "6e147360EXAMPLE",
            "comments": [
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
                    "clientRequestToken": "123Example",
                    "commentId": "ff30b348EXAMPLEb9aa670f",
                    "content": "Whoops - I meant to add this comment to the line,
 not the file, but I don't see how to delete it.",
                    "creationDate": 1508369768.142,
```

```
                    "deleted": false,
                    "CommentId": "123abc-EXAMPLE",
                    "lastModifiedDate": 1508369842.278,
                    "callerReactions": [],
                    "reactionCounts":
                    {
                        "SMILE" : 6,
                        "THUMBSUP" : 1
                    }
                },
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
                    "clientRequestToken": "123Example",
                    "commentId": "553b509bEXAMPLE56198325",
                    "content": "Can you add a test case for this?",
                    "creationDate": 1508369612.240,
                    "deleted": false,
                    "commentId": "456def-EXAMPLE",
                    "lastModifiedDate": 1508369612.240,
                    "callerReactions": [],
                    "reactionCounts":
                    {
                        "THUMBSUP" : 2
                    }
                }
            ],
            "location": {
                "filePath": "cl_sample.js",
                "filePosition": 1232,
                "relativeFileVersion": "after"
            },
            "repositoryName": "MyDemoRepo"
        }
    ],
    "nextToken": "exampleToken"
}
```

- Pour plus de détails sur l'API, reportez-vous GetCommentsForComparedCommità la section Référence des AWS CLI commandes.

`get-comments-for-pull-request`

L'exemple de code suivant montre comment utiliser`get-comments-for-pull-request`.

AWS CLI

Pour consulter les commentaires relatifs à une pull request

Cet exemple montre comment afficher les commentaires d'une pull request dans un référentiel nomméMyDemoRepo.

```
aws codecommit get-comments-for-pull-request \
    --repository-name MyDemoRepo \
    --before-commit-ID 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE
```

Sortie :

```
{
    "commentsForPullRequestData": [
        {
            "afterBlobId": "1f330709EXAMPLE",
            "afterCommitId": "5d036259EXAMPLE",
            "beforeBlobId": "80906a4cEXAMPLE",
            "beforeCommitId": "317f8570EXAMPLE",
            "comments": [
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
                    "clientRequestToken": "",
                    "commentId": "abcd1234EXAMPLEb5678efgh",
                    "content": "These don't appear to be used anywhere. Can we
 remove them?",
                    "creationDate": 1508369622.123,
                    "deleted": false,
                    "lastModifiedDate": 1508369622.123,
                    "callerReactions": [],
                    "reactionCounts":
                    {
                        "THUMBSUP" : 6,
                        "CONFUSED" : 1
                    }
                },
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
                    "clientRequestToken": "",
                    "commentId": "442b498bEXAMPLE5756813",
                    "content": "Good catch. I'll remove them.",
```

```
                    "creationDate": 1508369829.104,
                    "deleted": false,
                    "lastModifiedDate": 150836912.273,
                    "callerReactions": ["THUMBSUP"]
                    "reactionCounts":
                    {
                        "THUMBSUP" : 14
                    }
                }
            ],
            "location": {
                "filePath": "ahs_count.py",
                "filePosition": 367,
                "relativeFileVersion": "AFTER"
            },
            "repositoryName": "MyDemoRepo",
            "pullRequestId": "42"
        }
    ],
    "nextToken": "exampleToken"
}
```

- Pour plus de détails sur l'API, reportez-vous GetCommentsForPullRequestà la section Référence des AWS CLI commandes.

`get-commit`

L'exemple de code suivant montre comment utiliser`get-commit`.

AWS CLI

Pour afficher les informations relatives à un commit dans un référentiel

Cet exemple montre les détails d'un commit dont l'identifiant généré par le système est « 7e9fd3091thisisanexamplethisisanexample1 » dans un référentiel nommé « ». AWS CodeCommit MyDemoRepo

Commande :

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-
id 7e9fd3091thisisanexamplethisisanexample1
```

Sortie :

```
{
  "commit": {
      "additionalData": "",
      "committer": {
          "date": "1484167798 -0800",
          "name": "Mary Major",
          "email": "mary_major@example.com"
      },
      "author": {
          "date": "1484167798 -0800",
          "name": "Mary Major",
          "email": "mary_major@example.com"
      },
      "treeId": "347a3408thisisanexampletreeidexample",
      "parents": [
          "7aa87a031thisisanexamplethisisanexample1"
      ],
      "message": "Fix incorrect variable name"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetCommità la section Référence des AWS CLI commandes.

`get-differences`

L'exemple de code suivant montre comment utiliser`get-differences`.

AWS CLI

Pour obtenir des informations sur les différences entre un spécificateur de validation dans un référentiel

Cet exemple montre les informations de métadonnées d'affichage concernant les modifications entre deux spécificateurs de validation (branche, tag, HEAD ou autres références complètes, telles que commit IDs) dans un dossier renommé AWS CodeCommit du référentiel nommé MyDemoRepo. L'exemple inclut plusieurs options qui ne sont pas obligatoires, notamment --before-commit-specifier, --before-path et --after-path, afin d'illustrer de manière plus complète

comment vous pouvez utiliser ces options pour limiter les résultats. La réponse inclut les autorisations relatives au mode fichier.

Commande :

```
aws codecommit get-differences --repository-name MyDemoRepo --before-
commit-specifier 955bba12thisisanexamplethisisanexample --after-commit-
specifier 14a95463thisisanexamplethisisanexample --before-path tmp/example-folder --
after-path tmp/renamed-folder
```

Sortie :

```
{
  "differences": [
      {
          "afterBlob": {
              "path": "blob.txt",
              "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
              "mode": "100644"
          },
          "changeType": "M",
          "beforeBlob": {
              "path": "blob.txt",
              "blobId": "bf7fcf281thisisanexamplethisisanexample1",
              "mode": "100644"
          }
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDifferencesà la section Référence des AWS CLI commandes.

`get-file`

L'exemple de code suivant montre comment utiliser`get-file`.

AWS CLI

Pour obtenir le contenu codé en base 64 d'un fichier dans un dépôt AWS CodeCommit

L'`get-file`exemple suivant montre comment obtenir le contenu codé en base 64 d'un fichier nommé `README.md` à partir d'une branche nommée `main` dans un référentiel nommé. `MyDemoRepo`

```
aws codecommit get-file \
    --repository-name MyDemoRepo \
    --commit-specifier main \
    --file-path README.md
```

Sortie :

```
{
    "blobId":"559b44fEXAMPLE",
    "commitId":"c5709475EXAMPLE",
    "fileContent":"IyBQaHVzEXAMPLE",
    "filePath":"README.md",
    "fileMode":"NORMAL",
    "fileSize":1563
}
```

Pour plus d'informations, consultez GetFilele guide de référence des AWS CodeCommit API.

- Pour plus de détails sur l'API, reportez-vous GetFileà la section Référence des AWS CLI commandes.

`get-folder`

L'exemple de code suivant montre comment utiliser`get-folder`.

AWS CLI

Pour obtenir le contenu d'un dossier dans un AWS CodeCommit référentiel

L'`get-folder`exemple suivant montre comment obtenir le contenu d'un dossier de premier niveau à partir d'un référentiel nomméMyDemoRepo.

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

Sortie :

```
{
    "commitId":"c5709475EXAMPLE",
```

```
    "files":[
        {
            "absolutePath":".gitignore",
            "blobId":"74094e8bEXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":".gitignore"
        },
        {
            "absolutePath":"Gemfile",
            "blobId":"9ceb72f6EXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"Gemfile"
        },
        {
            "absolutePath":"Gemfile.lock",
            "blobId":"795c4a2aEXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"Gemfile.lock"
        },
        {
            "absolutePath":"LICENSE.txt",
            "blobId":"0c7932c8EXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"LICENSE.txt"
        },
        {
            "absolutePath":"README.md",
            "blobId":"559b44feEXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"README.md"
        }
    ],
    "folderPath":"",
    "subFolders":[
        {
            "absolutePath":"public",
            "relativePath":"public",
            "treeId":"d5e92ae3aEXAMPLE"
        },
        {
            "absolutePath":"tmp",
            "relativePath":"tmp",
            "treeId":"d564d0bcEXAMPLE"
        }
```

```
    ],
    "subModules":[],
    "symbolicLinks":[],
    "treeId":"7b3c4dadEXAMPLE"
}
```

Pour plus d'informations, consultez GetFolder le guide de référence des AWS CodeCommit API.

- Pour plus de détails sur l'API, reportez-vous GetFolderà la section Référence des AWS CLI commandes.

`get-merge-commit`

L'exemple de code suivant montre comment utiliser`get-merge-commit`.

AWS CLI

Pour obtenir des informations détaillées sur un commit de fusion

L'`get-merge-commit`exemple suivant affiche les détails d'un commit de fusion pour la branche source nommée `bugfix-bug1234` avec une branche de destination nommée `main` dans un référentiel nommé`MyDemoRepo`.

```
aws codecommit get-merge-commit \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "sourceCommitId": "c5709475EXAMPLE",
    "destinationCommitId": "317f8570EXAMPLE",
    "baseCommitId": "fb12a539EXAMPLE",
    "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

Pour plus d'informations, voir Afficher les détails des validations dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetMergeCommità la section Référence des AWS CLI commandes.

`get-merge-conflicts`

L'exemple de code suivant montre comment utiliser`get-merge-conflicts`.

AWS CLI

Pour voir s'il existe des conflits de fusion pour une pull request

L'`get-merge-conflicts`exemple suivant indique s'il existe des conflits de fusion entre l'extrémité d'une branche source nommée `feature-randomizationfeature` et une branche de destination nommée « main » dans un référentiel nomméMyDemoRepo.

```
aws codecommit get-merge-conflicts \
    --repository-name MyDemoRepo \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE
```

Sortie :

```
{
    "mergeable": false,
    "destinationCommitId": "86958e0aEXAMPLE",
    "sourceCommitId": "6ccd57fdEXAMPLE",
    "baseCommitId": "767b6958EXAMPLE",
    "conflictMetadataList": [
        {
            "filePath": "readme.md",
            "fileSizes": {
                "source": 139,
                "destination": 230,
                "base": 85
            },
            "fileModes": {
                "source": "NORMAL",
                "destination": "NORMAL",
                "base": "NORMAL"
            },
            "objectTypes": {
                "source": "FILE",
                "destination": "FILE",
                "base": "FILE"
            },
```

```
            "numberOfConflicts": 1,
            "isBinaryFile": {
                "source": false,
                "destination": false,
                "base": false
            },
            "contentConflict": true,
            "fileModeConflict": false,
            "objectTypeConflict": false,
            "mergeOperations": {
                "source": "M",
                "destination": "M"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetMergeConflictsà la section Référence des AWS CLI commandes.

`get-merge-options`

L'exemple de code suivant montre comment utiliser`get-merge-options`.

AWS CLI

Pour obtenir des informations sur les options de fusion disponibles pour fusionner deux branches spécifiées

L'`get-merge-options`exemple suivant détermine les options de fusion disponibles pour fusionner une branche source nommée `bugfix-bug1234` avec une branche de destination nommée `main` dans un référentiel nommé`MyDemoRepo`.

```
aws codecommit get-merge-options \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
    --repository-name MyDemoRepo
```

Sortie :

```
{
```

```
    "mergeOptions": [
        "FAST_FORWARD_MERGE",
        "SQUASH_MERGE",
        "THREE_WAY_MERGE"
    ],
    "sourceCommitId": "18059494EXAMPLE",
    "destinationCommitId": "ffd3311dEXAMPLE",
    "baseCommitId": "ffd3311dEXAMPLE"
}
```

Pour plus d'informations, consultez la section Résoudre les conflits liés à une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetMergeOptionsà la section Référence des AWS CLI commandes.

`get-pull-request-approval-states`

L'exemple de code suivant montre comment utiliser`get-pull-request-approval-states`.

AWS CLI

Pour consulter les approbations sur une pull request

L'`get-pull-request-approval-states`exemple suivant renvoie les approbations pour la pull request spécifiée.

```
aws codecommit get-pull-request-approval-states \
    --pull-request-id 8 \
    --revision-id 9f29d167EXAMPLE
```

Sortie :

```
{
    "approvals": [
        {
            "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
            "approvalState": "APPROVE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les pull requests dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetPullRequestApprovalStatesà la section Référence des AWS CLI commandes.

`get-pull-request-override-state`

L'exemple de code suivant montre comment utiliser`get-pull-request-override-state`.

AWS CLI

Pour obtenir des informations sur le statut de dérogation d'une pull request

L'`get-pull-request-override-state`exemple suivant renvoie l'état de remplacement pour la pull request spécifiée. Dans cet exemple, les règles d'approbation de la pull request ont été annulées par une utilisatrice nommée Mary Major, de sorte que la sortie renvoie une valeur de. `true` :

```
aws codecommit get-pull-request-override-state \
    --pull-request-id 34  \
    --revision-id 9f29d167EXAMPLE
```

Sortie :

```
{
    "overridden": true,
    "overrider": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

Pour plus d'informations, consultez la section Remplacer les règles d'approbation relatives à une pull request dans le guide de l'AWS CodeCommit utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetPullRequestOverrideStateà la section Référence des AWS CLI commandes.

`get-pull-request`

L'exemple de code suivant montre comment utiliser`get-pull-request`.

AWS CLI

Pour afficher les détails d'une pull request

Cet exemple montre comment afficher les informations relatives à une pull request avec l'ID de27.

```
aws codecommit get-pull-request \
    --pull-request-id 27
```

Sortie :

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "lastActivityDate": 1562619583.565,
        "pullRequestTargets": [
            {
                "sourceCommit": "ca45e279EXAMPLE",
                "sourceReference": "refs/heads/bugfix-1234",
                "mergeBase": "a99f5ddbEXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false
                },
                "destinationCommit": "2abfc6beEXAMPLE",
                "repositoryName": "MyDemoRepo"
            }
        ],
        "revisionId": "e47def21EXAMPLE",
        "title": "Quick fix for bug 1234",
        "authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
```

```
        "clientRequestToken": "d8d7612e-EXAMPLE",
        "creationDate": 1562619583.565,
        "pullRequestId": "27",
        "pullRequestStatus": "OPEN"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPullRequestà la section Référence des AWS CLI commandes.

`get-repository-triggers`

L'exemple de code suivant montre comment utiliser`get-repository-triggers`.

AWS CLI

Pour obtenir des informations sur les déclencheurs dans un référentiel

Cet exemple montre des détails sur les déclencheurs configurés pour un AWS CodeCommit référentiel nomméMyDemoRepo.

```
aws codecommit get-repository-triggers \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "configurationId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    "triggers": [
        {
            "destinationArn": "arn:aws:sns:us-
east-1:111111111111:MyCodeCommitTopic",
            "branches": [
                "main",
                "preprod"
            ],
            "name": "MyFirstTrigger",
            "customData": "",
            "events": [
                "all"
            ]
```

```
        },
        {
            "destinationArn": "arn:aws:lambda:us-
east-1:111111111111:function:MyCodeCommitPythonFunction",
            "branches": [],
            "name": "MySecondTrigger",
            "customData": "EXAMPLE",
            "events": [
                "all"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRepositoryTriggersà la section Référence des AWS CLI commandes.

`get-repository`

L'exemple de code suivant montre comment utiliser`get-repository`.

AWS CLI

Pour obtenir des informations sur un référentiel

Cet exemple montre les détails d'un AWS CodeCommit dépôt.

```
aws codecommit get-repository \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "repositoryMetadata": {
        "creationDate": 1429203623.625,
        "defaultBranch": "main",
        "repositoryName": "MyDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/v1/
repos/MyDemoRepo",
        "lastModifiedDate": 1430783812.0869999,
        "repositoryDescription": "My demonstration repository",
```

```
        "cloneUrlHttp": "https://codecommit.us-east-1.amazonaws.com/v1/repos/
MyDemoRepo",
        "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-1:80398EXAMPLE:MyDemoRepo
        "accountId": "111111111111"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetRepositoryà la section Référence des AWS CLI commandes.

`list-approval-rule-templates`

L'exemple de code suivant montre comment utiliser`list-approval-rule-templates`.

AWS CLI

Pour répertorier tous les modèles de règles d'approbation d'une AWS région

L'`list-approval-rule-templates`exemple suivant répertorie tous les modèles de règles d'approbation dans la région spécifiée. Si aucune AWS région n'est spécifiée en tant que paramètre, la commande renvoie des modèles de règles d'approbation pour la région spécifiée dans le profil AWS CLI utilisé pour exécuter la commande.

```
aws codecommit list-approval-rule-templates \
    --region us-east-2
```

Sortie :

```
{
    "approvalRuleTemplateNames": [
        "2-approver-rule-for-main",
        "1-approver-rule-for-all-pull-requests"
    ]
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListApprovalRuleTemplatesà la section Référence des AWS CLI commandes.

`list-associated-approval-rule-templates-for-repository`

L'exemple de code suivant montre comment utiliser`list-associated-approval-rule-templates-for-repository`.

AWS CLI

Pour répertorier tous les modèles associés à un référentiel

L'`list-associated-approval-rule-templates-for-repository`exemple suivant répertorie tous les modèles de règles d'approbation associés à un référentiel nomméMyDemoRepo.

```
aws codecommit list-associated-approval-rule-templates-for-repository \
    --repository-name MyDemoRepo
```

Sortie :

```
{
  "approvalRuleTemplateNames": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListAssociatedApprovalRuleTemplatesForRepositoryà la section Référence des AWS CLI commandes.

`list-branches`

L'exemple de code suivant montre comment utiliser`list-branches`.

AWS CLI

Pour consulter la liste des noms de succursales

Cet exemple répertorie tous les noms de branches d'un AWS CodeCommit référentiel.

```
aws codecommit list-branches \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "branches": [
        "MyNewBranch",
        "main"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListBranchesà la section Référence des AWS CLI commandes.

`list-pull-requests`

L'exemple de code suivant montre comment utiliser`list-pull-requests`.

AWS CLI

Pour afficher la liste des pull requests dans un référentiel

Cet exemple montre comment répertorier les pull requests créées par un utilisateur IAM avec l'ARN « arn:aws:iam : :111111111111:user/LI_JUAN » et le statut « CLOSED » dans un référentiel nommé « » : AWS CodeCommit MyDemoRepo

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/
Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

Sortie :

```
{
 "nextToken": "",
 "pullRequestIds": ["2","12","16","22","23","35","30","39","47"]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPullRequestsà la section Référence des AWS CLI commandes.

`list-repositories-for-approval-rule-template`

L'exemple de code suivant montre comment utiliser`list-repositories-for-approval-rule-template`.

AWS CLI

Pour répertorier tous les référentiels associés à un modèle

L'`list-repositories-for-approval-rule-template`exemple suivant répertorie tous les référentiels associés au modèle de règle d'approbation spécifié.

```
aws codecommit list-repositories-for-approval-rule-template \
    --approval-rule-template-name 2-approver-rule-for-main
```

Sortie :

```
{
    "repositoryNames": [
        "MyDemoRepo",
        "MyClonedRepo"
    ]
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListRepositoriesForApprovalRuleTemplateà la section Référence des AWS CLI commandes.

`list-repositories`

L'exemple de code suivant montre comment utiliser`list-repositories`.

AWS CLI

Pour afficher la liste des référentiels

Cet exemple répertorie tous les AWS CodeCommit référentiels associés au AWS compte de l'utilisateur.

Commande :

```
aws codecommit list-repositories
```

Sortie :

```
{
  "repositories": [
      {
         "repositoryName": "MyDemoRepo"
         "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
      },
      {
         "repositoryName": "MyOtherDemoRepo"
         "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListRepositoriesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour afficher les AWS balises d'un référentiel

L'`list-tags-for-resource`exemple suivant répertorie les clés de balise et les valeurs de balise pour le référentiel spécifié.

```
aws codecommit list-tags-for-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

Sortie :

```
{
    "tags": {
        "Status": "Secret",
        "Team": "Saanvi"
```

```
    }
}
```

Pour plus d'informations, voir Afficher les balises d'un référentiel dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`merge-branches-by-fast-forward`

L'exemple de code suivant montre comment utiliser`merge-branches-by-fast-forward`.

AWS CLI

Pour fusionner deux succursales à l'aide de la stratégie de fusion accélérée

L'`merge-branches-by-fast-forward`exemple suivant fusionne la branche source spécifiée avec la branche de destination spécifiée dans un référentiel nommé`MyDemoRepo`.

```
aws codecommit merge-branches-by-fast-forward \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier bugfix-bug1233 \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

Pour plus d'informations, consultez la section Comparer et fusionner des branches dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous MergeBranchesByFastForwardà la section Référence des AWS CLI commandes.

`merge-branches-by-squash`

L'exemple de code suivant montre comment utiliser`merge-branches-by-squash`.

AWS CLI

Pour fusionner deux branches à l'aide de la stratégie de fusion Squash

L'`merge-branches-by-squash`exemple suivant fusionne la branche source spécifiée avec la branche de destination spécifiée dans un référentiel nomméMyDemoRepo.

```
aws codecommit merge-branches-by-squash \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier bugfix-bug1233 \
    --author-name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "Merging two fix branches to prepare for a general patch." \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

Pour plus d'informations, consultez la section Comparer et fusionner des branches dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous MergeBranchesBySquashà la section Référence des AWS CLI commandes.

`merge-branches-by-three-way`

L'exemple de code suivant montre comment utiliser`merge-branches-by-three-way`.

AWS CLI

Pour fusionner deux succursales à l'aide de la stratégie de fusion à trois

L'`merge-branches-by-three-way`exemple suivant fusionne la branche source spécifiée avec la branche de destination spécifiée dans un référentiel nomméMyDemoRepo.

```
aws codecommit merge-branches-by-three-way \
    --source-commit-specifier main \
```

```
    --destination-commit-specifier bugfix-bug1234 \
    --author-name "Jorge Souza" --email "jorge_souza@example.com" \
    --commit-message "Merging changes from main to bugfix branch before additional
 testing." \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

Pour plus d'informations, consultez la section Comparer et fusionner des branches dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous MergeBranchesByThreeWayà la section Référence des AWS CLI commandes.

`merge-pull-request-by-fast-forward`

L'exemple de code suivant montre comment utiliser`merge-pull-request-by-fast-forward`.

AWS CLI

Pour fusionner et fermer une pull request

Cet exemple montre comment fusionner et fermer une pull request ayant l'ID « 47 » et l'ID de validation source « 99132ab0Example » dans un référentiel nommé. MyDemoRepo

```
aws codecommit merge-pull-request-by-fast-forward \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo
```

Sortie :

```
{
    "pullRequest": {
        "approvalRules": [
            {
```

```
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "I want one approver for this pull request",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.142,
        "description": "Review the latest changes and updates to the global
 variables",
        "lastActivityDate": 1508887223.155,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": true,
                    "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

Pour plus d'informations, consultez la section Fusionner une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous MergePullRequestByFastForwardà la section Référence des AWS CLI commandes.

merge-pull-request-by-squash

L'exemple de code suivant montre comment utilisermerge-pull-request-by-squash.

AWS CLI

Pour fusionner une pull request à l'aide de la stratégie de fusion Squash

L'merge-pull-request-by-squashexemple suivant fusionne et ferme la pull request spécifiée en utilisant la stratégie de résolution de conflits d'ACCEPT_SOURCE dans un référentiel nommé. MyDemoRepo

```
aws codecommit merge-pull-request-by-squash \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo \
    --conflict-detail-level LINE_LEVEL \
    --conflict-resolution-strategy ACCEPT_SOURCE \
    --name "Jorge Souza" --email "jorge_souza@example.com" \
    --commit-message "Merging pull request 47 by squash and accepting source in
 merge conflicts"
```

Sortie :

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
```

```
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.142,
        "description": "Review the latest changes and updates to the global
 variables",
        "lastActivityDate": 1508887223.155,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": true,
                    "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

Pour plus d'informations, consultez la section Fusionner une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous MergePullRequestBySquashà la section Référence des AWS CLI commandes.

`merge-pull-request-by-three-way`

L'exemple de code suivant montre comment utiliser`merge-pull-request-by-three-way`.

AWS CLI

Pour fusionner une pull request à l'aide de la stratégie de fusion à trois voies

L'`merge-pull-request-by-three-way`exemple suivant fusionne et ferme la pull request spécifiée à l'aide des options par défaut relatives au détail des conflits et à la stratégie de résolution des conflits dans un référentiel nomméMyDemoRepo.

```
aws codecommit merge-pull-request-by-three-way \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo \
    --name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "Merging pull request 47 by three-way with default options"
```

Sortie :


```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.142,
        "description": "Review the latest changes and updates to the global
 variables",
        "lastActivityDate": 1508887223.155,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
```

```
                    "isMerged": true,
                    "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

Pour plus d'informations, consultez la section Fusionner une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous MergePullRequestByThreeWayà la section Référence des AWS CLI commandes.

`override-pull-request-approval-rules`

L'exemple de code suivant montre comment utiliser`override-pull-request-approval-rules`.

AWS CLI

Pour contourner les exigences des règles d'approbation relatives à une pull request

L'`override-pull-request-approval-rules`exemple suivant remplace les règles d'approbation relatives à la pull request spécifiée. Pour annuler une dérogation à la place, définissez la valeur du `--override-status` paramètre sur. `REVOKE`

```
aws codecommit override-pull-request-approval-rules \
    --pull-request-id 34  \
    --revision-id 927df8d8EXAMPLE \
    --override-status OVERRIDE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Remplacer les règles d'approbation relatives à une pull request dans le guide de l'AWS CodeCommit utilisateur.

- Pour plus de détails sur l'API, reportez-vous OverridePullRequestApprovalRulesà la section Référence des AWS CLI commandes.

`post-comment-for-compared-commit`

L'exemple de code suivant montre comment utiliser`post-comment-for-compared-commit`.

AWS CLI

Pour créer un commentaire sur un commit

Cet exemple montre comment ajouter le commentaire `"Can you add a test case for this?"` sur la modification apportée au `cl_sample.js` fichier lors de la comparaison entre deux validations dans un référentiel nomméMyDemoRepo.

```
aws codecommit post-comment-for-compared-commit \
    --repository-name MyDemoRepo \
    --before-commit-id 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE \
    --client-request-token 123Example \
    --content "Can you add a test case for this?" \
    --location filePath=cl_sample.js,filePosition=1232,relativeFileVersion=AFTER
```

Sortie :

```
{
    "afterBlobId": "1f330709EXAMPLE",
    "afterCommitId": "317f8570EXAMPLE",
    "beforeBlobId": "80906a4cEXAMPLE",
    "beforeCommitId": "6e147360EXAMPLE",
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "",
        "commentId": "553b509bEXAMPLE56198325",
        "content": "Can you add a test case for this?",
        "creationDate": 1508369612.203,
        "deleted": false,
        "commentId": "abc123-EXAMPLE",
        "lastModifiedDate": 1508369612.203,
        "callerReactions": [],
        "reactionCounts": []
    },
    "location": {
        "filePath": "cl_sample.js",
        "filePosition": 1232,
```

```
        "relativeFileVersion": "AFTER"
    ,
    "repositoryName": "MyDemoRepo"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PostCommentForComparedCommitàla section Référence des AWS CLI commandes.

`post-comment-for-pull-request`

L'exemple de code suivant montre comment utiliser`post-comment-for-pull-request`.

AWS CLI

Pour ajouter un commentaire à une pull request

L'`post-comment-for-pull-request`exemple suivant ajoute le commentaire « Ils ne semblent être utilisés nulle part. Can we remove them? » sur la modification du `ahs_count.py` fichier dans une pull request avec l'ID de 47 dans un dépôt nomméMyDemoRepo.

```
aws codecommit post-comment-for-pull-request \
    --pull-request-id "47" \
    --repository-name MyDemoRepo \
    --before-commit-id 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE \
    --client-request-token 123Example \
    --content "These don't appear to be used anywhere. Can we remove them?" \
    --location filePath=ahs_count.py,filePosition=367,relativeFileVersion=AFTER
```

Sortie :

```
{
     "afterBlobId": "1f330709EXAMPLE",
     "afterCommitId": "5d036259EXAMPLE",
     "beforeBlobId": "80906a4cEXAMPLE",
     "beforeCommitId": "317f8570EXAMPLE",
     "comment": {
            "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
            "clientRequestToken": "123Example",
            "commentId": "abcd1234EXAMPLEb5678efgh",
```

```
            "content": "These don't appear to be used anywhere. Can we remove
 them?",
            "creationDate": 1508369622.123,
            "deleted": false,
            "CommentId": "",
            "lastModifiedDate": 1508369622.123,
            "callerReactions": [],
            "reactionCounts": []
        },
        "location": {
            "filePath": "ahs_count.py",
            "filePosition": 367,
            "relativeFileVersion": "AFTER"
         },
     "repositoryName": "MyDemoRepo",
     "pullRequestId": "47"
}
```

- Pour plus de détails sur l'API, reportez-vous PostCommentForPullRequestà la section Référence des AWS CLI commandes.

`post-comment-reply`

L'exemple de code suivant montre comment utiliser`post-comment-reply`.

AWS CLI

Pour répondre à un commentaire sur un commit ou dans une pull request

Cet exemple montre comment ajouter la réponse `"Good catch. I'll remove them."` au commentaire avec l'ID généré par le système de. `abcd1234EXAMPLEb5678efgh`

```
aws codecommit post-comment-reply \
    --in-reply-to abcd1234EXAMPLEb5678efgh \
    --content "Good catch. I'll remove them." \
    --client-request-token 123Example
```

Sortie :

```
{
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
```

```
        "clientRequestToken": "123Example",
        "commentId": "442b498bEXAMPLE5756813",
        "content": "Good catch. I'll remove them.",
        "creationDate": 1508369829.136,
        "deleted": false,
        "CommentId": "abcd1234EXAMPLEb5678efgh",
        "lastModifiedDate": 150836912.221,
        "callerReactions": [],
        "reactionCounts": []
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PostCommentReplyà la section Référence des AWS CLI commandes.

`put-comment-reaction`

L'exemple de code suivant montre comment utiliser`put-comment-reaction`.

AWS CLI

Pour répondre à un commentaire sur un commit avec un emoji

L'`put-comment-reaction`exemple suivant répond à un commentaire avec l'ID de `abcd1234EXAMPLEb5678efgh` avec une valeur de réaction emoji de`:thumbsup:`.

```
aws codecommit put-comment-reaction \
    --comment-id abcd1234EXAMPLEb5678efgh \
    --reaction-value :thumbsup:
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Commenter un commit AWS CodeCommit dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous PutCommentReactionà la section Référence des AWS CLI commandes.

`put-file`

L'exemple de code suivant montre comment utiliser`put-file`.

AWS CLI

Pour ajouter un fichier à un référentiel

L'`put-file`exemple suivant ajoute un fichier nommé « ExampleSolution .py » à un référentiel nommé « » dans une branche nommée « MyDemoRepo feature-randomizationfeature » dont le dernier commit porte l'ID « 4C925148Example ».

```
aws codecommit put-file \
    --repository-name MyDemoRepo \
    --branch-name feature-randomizationfeature \
    --file-content file://MyDirectory/ExampleSolution.py \
    --file-path /solutions/ExampleSolution.py \
    --parent-commit-id 4c925148EXAMPLE \
    --name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "I added a third randomization routine."
```

Sortie :

```
{
    "blobId": "2eb4af3bEXAMPLE",
    "commitId": "317f8570EXAMPLE",
    "treeId": "347a3408EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous PutFileà la section Référence des AWS CLI commandes.

`put-repository-triggers`

L'exemple de code suivant montre comment utiliser`put-repository-triggers`.

AWS CLI

Pour ajouter ou mettre à jour un déclencheur dans un référentiel

Cet exemple montre comment mettre à jour les déclencheurs nommés « MyFirstTrigger » et « MySecondTrigger » à l'aide d'un fichier JSON déjà créé (ici nommé MyTriggers .json) qui contient la structure de tous les déclencheurs d'un référentiel nommé. MyDemoRepo Pour savoir

comment obtenir le JSON pour les déclencheurs existants, consultez la get-repository-triggers commande.

```
aws codecommit put-repository-triggers \
    --repository-name MyDemoRepo file://MyTriggers.json
```

Contenu de `MyTriggers.json` :

```
{
    "repositoryName": "MyDemoRepo",
    "triggers": [
        {
            "destinationArn": "arn:aws:sns:us-
east-1:80398EXAMPLE:MyCodeCommitTopic",
            "branches": [
                "main",
                "preprod"
            ],
            "name": "MyFirstTrigger",
            "customData": "",
            "events": [
                "all"
            ]
        },
        {
            "destinationArn": "arn:aws:lambda:us-
east-1:111111111111:function:MyCodeCommitPythonFunction",
            "branches": [],
            "name": "MySecondTrigger",
            "customData": "EXAMPLE",
            "events": [
                "all"
            ]
        }
    ]
}
```

Sortie :

```
{
    "configurationId": "6fa51cd8-35c1-EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous PutRepositoryTriggersà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des AWS balises à un référentiel existant

L'`tag-resource`exemple suivant balise le référentiel spécifié avec deux balises.

```
aws codecommit tag-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
    --tags Status=Secret,Team=Saanvi
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Ajouter une balise à un référentiel dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`test-repository-triggers`

L'exemple de code suivant montre comment utiliser`test-repository-triggers`.

AWS CLI

Pour tester les déclencheurs dans un référentiel

Cet exemple montre comment tester un déclencheur nommé « MyFirstTrigger » dans un AWS CodeCommit référentiel nommé MyDemoRepo. Dans cet exemple, les événements du référentiel déclenchent des notifications provenant d'une rubrique Amazon Simple Notification Service (Amazon SNS).

Commande :

```
aws codecommit test-repository-triggers --repository-name MyDemoRepo
 --triggers name=MyFirstTrigger,destinationArn=arn:aws:sns:us-
east-1:111111111111:MyCodeCommitTopic,branches=mainline,preprod,events=all
```

Sortie :

```
{
  "successfulExecutions": [
      "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- Pour plus de détails sur l'API, reportez-vous TestRepositoryTriggersà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des AWS balises d'un référentiel

L'`untag-resource`exemple suivant supprime la balise avec la clé spécifiée du référentiel nommé`MyDemoRepo`.

```
aws codecommit untag-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
    --tag-keys Status
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer une étiquette d'un référentiel dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

update-approval-rule-template-content

L'exemple de code suivant montre comment utiliserupdate-approval-rule-template-content.

AWS CLI

Pour mettre à jour le contenu d'un modèle de règle d'approbation

L'update-approval-rule-template-contentexemple suivant modifie le contenu du modèle de règle d'approbation spécifié afin de redéfinir le pool d'approbation pour les utilisateurs qui assument le rôle deCodeCommitReview.

```
aws codecommit update-approval-rule-template-content \
    --approval-rule-template-name 1-approver-rule  \
    --new-rule-content "{\"Version\": \"2018-11-08\",\"DestinationReferences
\": [\"refs/heads/main\"],\"Statements\": [{\"Type\": \"Approvers
\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

Sortie :

```
{
    "approvalRuleTemplate": {
        "creationDate": 1571352720.773,
        "approvalRuleTemplateDescription": "Requires 1 approval for all pull
 requests from the CodeCommitReview pool",
        "lastModifiedDate": 1571358728.41,
        "approvalRuleTemplateId": "41de97b7-EXAMPLE",
        "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
        "ruleContentSha256": "2f6c21a5EXAMPLE",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
    }
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateApprovalRuleTemplateContentà la section Référence des AWS CLI commandes.

update-approval-rule-template-description

L'exemple de code suivant montre comment utiliserupdate-approval-rule-template-description.

AWS CLI

Pour mettre à jour la description d'un modèle de règle d'approbation

L'update-approval-rule-template-descriptionexemple suivant change la description du modèle de règle d'approbation spécifié enRequires 1 approval for all pull requests from the CodeCommitReview pool. :

```
aws codecommit update-approval-rule-template-description \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests  \
    --approval-rule-template-description "Requires 1 approval for all pull requests
 from the CodeCommitReview pool"
```

Sortie :

```
{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull requests
 from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateApprovalRuleTemplateDescriptionà la section Référence des AWS CLI commandes.

update-approval-rule-template-name

L'exemple de code suivant montre comment utiliserupdate-approval-rule-template-name.

AWS CLI

Pour mettre à jour le nom d'un modèle de règle d'approbation

L'update-approval-rule-template-nameexemple suivant change le nom d'un modèle de règle d'approbation en 1- approver-rule-for-all -pull-requests`. 1-approver-rule

```
aws codecommit update-approval-rule-template-name \
    --old-approval-rule-template-name 1-approver-rule  \
    --new-approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Sortie :

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedDate": 1571358241.619,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "creationDate": 1571352720.773,
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "approvalRuleTemplateDescription": "All pull requests must be approved by one
 developer on the team.",
    "ruleContentSha256": "2f6c21a5cEXAMPLE"
  }
}
```

Pour plus d'informations, consultez la section Gérer les modèles de règles d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateApprovalRuleTemplateNameà la section Référence des AWS CLI commandes.

update-comment

L'exemple de code suivant montre comment utiliserupdate-comment.

AWS CLI

Pour mettre à jour un commentaire sur un commit

Cet exemple montre comment ajouter du contenu `"Fixed as requested. I'll update the pull request."` à un commentaire dont l'ID est`442b498bEXAMPLE5756813`.

```
aws codecommit update-comment \
    --comment-id 442b498bEXAMPLE5756813 \
    --content "Fixed as requested. I'll update the pull request."
```

Sortie :

```
{
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "",
        "commentId": "442b498bEXAMPLE5756813",
        "content": "Fixed as requested. I'll update the pull request.",
        "creationDate": 1508369929.783,
        "deleted": false,
        "lastModifiedDate": 1508369929.287,
        "callerReactions": [],
        "reactionCounts":
            {
                "THUMBSUP" : 2
            }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateCommentà la section Référence des AWS CLI commandes.

`update-default-branch`

L'exemple de code suivant montre comment utiliser`update-default-branch`.

AWS CLI

Pour modifier la branche par défaut d'un référentiel

Cet exemple modifie la branche par défaut d'un AWS CodeCommit référentiel. Cette commande génère une sortie uniquement en cas d'erreurs.

Commande :

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-
name MyNewBranch
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous UpdateDefaultBranchà la section Référence des AWS CLI commandes.

`update-pull-request-approval-rule-content`

L'exemple de code suivant montre comment utiliser`update-pull-request-approval-rule-content`.

AWS CLI

Pour modifier une règle d'approbation pour une pull request

L'`update-pull-request-approval-rule-content`exemple suivant met à jour la règle d'approbation qu'elle a spécifiée pour exiger l'approbation d'un utilisateur à partir d'un pool d'approbation qui inclut n'importe quel utilisateur IAM dans le `123456789012` AWS compte.

```
aws codecommit update-pull-request-approval-rule-content \
    --pull-request-id 27  \
    --approval-rule-name "Require two approved approvers" \
    --approval-rule-content "{Version: 2018-11-08, Statements: [{Type:
 \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
[\"CodeCommitApprovers:123456789012:user/*\"]}]}"
```

Sortie :

```
{
    "approvalRule": {
```

```
        "approvalRuleContent": "{Version: 2018-11-08, Statements:
 [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
[\"CodeCommitApprovers:123456789012:user/*\"]}]}",
        "approvalRuleId": "aac33506-EXAMPLE",
        "originApprovalRuleTemplate": {},
        "creationDate": 1570752871.932,
        "lastModifiedDate": 1570754058.333,
        "approvalRuleName": Require two approved approvers",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "cd93921cEXAMPLE",
    }
}
```

Pour plus d'informations, voir Modifier ou supprimer une règle d'approbation dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdatePullRequestApprovalRuleContentà la section Référence des AWS CLI commandes.

`update-pull-request-approval-state`

L'exemple de code suivant montre comment utiliser`update-pull-request-approval-state`.

AWS CLI

Pour approuver ou révoquer l'approbation d'une pull request

L'`update-pull-request-approval-state`exemple suivant approuve une pull request avec l'ID 27 et l'ID de révision de`9f29d167EXAMPLE`. Si vous souhaitez plutôt révoquer l'approbation, définissez la valeur du `--approval-state` paramètre sur`REVOKE`.

```
aws codecommit update-pull-request-approval-state \
    --pull-request-id 27  \
    --revision-id 9f29d167EXAMPLE  \
    --approval-state "APPROVE"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Révision d'une pull request dans le guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdatePullRequestApprovalStateà la section Référence des AWS CLI commandes.

update-pull-request-description

L'exemple de code suivant montre comment utiliserupdate-pull-request-description.

AWS CLI

Pour modifier la description d'une pull request

Cet exemple montre comment modifier la description d'une pull request avec l'ID de47.

```
aws codecommit update-pull-request-description \
    --pull-request-id 47 \
    --description "Updated the pull request to remove unused global variable."
```

Sortie :

```
{
    "pullRequest": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.155,
        "description": "Updated the pull request to remove unused global variable.",
        "lastActivityDate": 1508372423.204,
        "pullRequestId": "47",
        "pullRequestStatus": "OPEN",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdatePullRequestDescriptionà la section Référence des AWS CLI commandes.

update-pull-request-status

L'exemple de code suivant montre comment utiliserupdate-pull-request-status.

AWS CLI

Pour modifier le statut d'une pull request

Cet exemple montre comment modifier le statut d'une pull request avec l'ID de 42 en un statut de CLOSED dans un AWS CodeCommit référentiel nomméMyDemoRepo.

```
aws codecommit update-pull-request-status \
    --pull-request-id 42 \
    --pull-request-status CLOSED
```

Sortie :

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approvers-needed-for-this-change",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.165,
        "description": "Updated the pull request to remove unused global variable.",
        "lastActivityDate": 1508372423.12,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
```

```
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdatePullRequestStatusà la section Référence des AWS CLI commandes.

`update-pull-request-title`

L'exemple de code suivant montre comment utiliser`update-pull-request-title`.

AWS CLI

Pour modifier le titre d'une pull request

Cet exemple montre comment modifier le titre d'une pull request avec l'ID de47.

```
aws codecommit update-pull-request-title \
    --pull-request-id 47 \
    --title "Consolidation of global variables - updated review"
```

Sortie :

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
```

```
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd8b26gr-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.12,
        "description": "Review the latest changes and updates to the global
 variables. I have updated this request with some changes, including removing some
 unused variables.",
        "lastActivityDate": 1508372657.188,
        "pullRequestId": "47",
        "pullRequestStatus": "OPEN",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables - updated review"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdatePullRequestTitleà la section Référence des AWS CLI commandes.

`update-repository-description`

L'exemple de code suivant montre comment utiliser`update-repository-description`.

AWS CLI

Pour modifier la description d'un référentiel

Cet exemple modifie la description d'un AWS CodeCommit référentiel. Cette commande génère une sortie uniquement en cas d'erreurs.

Commande :

```
aws codecommit update-repository-description --repository-name MyDemoRepo --
repository-description "This description was changed"
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous UpdateRepositoryDescriptionà la section Référence des AWS CLI commandes.

`update-repository-name`

L'exemple de code suivant montre comment utiliser`update-repository-name`.

AWS CLI

Pour modifier le nom d'un dépôt

Cet exemple modifie le nom d'un AWS CodeCommit dépôt. Cette commande génère une sortie uniquement en cas d'erreurs. La modification du nom du AWS CodeCommit référentiel modifiera les protocoles SSH et HTTPS URLs dont les utilisateurs ont besoin pour se connecter au référentiel. Les utilisateurs ne pourront pas se connecter à ce référentiel tant qu'ils n'auront pas mis à jour leurs paramètres de connexion. De plus, étant donné que l'ARN du référentiel va changer, la modification du nom du référentiel invalidera toutes les politiques utilisateur IAM qui s'appuient sur l'ARN de ce référentiel.

Commande :

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-
name MyRenamedDemoRepo
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous UpdateRepositoryNameà la section Référence des AWS CLI commandes.

CodeDeploy exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CodeDeploy.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-on-premises-instances`

L'exemple de code suivant montre comment utiliser`add-tags-to-on-premises-instances`.

AWS CLI

Pour ajouter des balises aux instances locales

L'`add-tags-to-on-premises-instances`exemple suivant associe AWS CodeDeploy la même balise d'instance locale à deux instances locales. Il n'enregistre pas les instances locales auprès de. AWS CodeDeploy

```
aws deploy add-tags-to-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AddTagsToOnPremisesInstancesà la section Référence des AWS CLI commandes.

`batch-get-application-revisions`

L'exemple de code suivant montre comment utiliser`batch-get-application-revisions`.

AWS CLI

Pour récupérer des informations sur les révisions des applications

L'`batch-get-application-revisions`exemple suivant récupère les informations relatives à la révision spécifiée stockée dans un GitHub référentiel.

```
aws deploy batch-get-application-revisions \
    --application-name my-codedeploy-application \
    --revisions "[{\"gitHubLocation\": {\"commitId\":
 \"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\",\"repository\": \"my-github-token/my-
repository\"},\"revisionType\": \"GitHub\"}]"
```

Sortie :

```
{
    "revisions": [
        {
            "genericRevisionInfo": {
                "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
                "lastUsedTime": 1556912355.884,
                "registerTime": 1556912355.884,
                "firstUsedTime": 1556912355.884,
                "deploymentGroups": []
            },
            "revisionLocation": {
                "revisionType": "GitHub",
                "gitHubLocation": {
                    "commitId": "fa85936EXAMPLEa31736c051f10d77297EXAMPLE",
                    "repository": "my-github-token/my-repository"
                }
            }
        }
```

```
    ],
    "applicationName": "my-codedeploy-application",
    "errorMessage": ""
}
```

Pour plus d'informations, consultez BatchGetApplicationRevisions dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous BatchGetApplicationRevisionsà la section Référence des AWS CLI commandes.

`batch-get-applications`

L'exemple de code suivant montre comment utiliser`batch-get-applications`.

AWS CLI

Pour obtenir des informations sur plusieurs applications

L'`batch-get-applications`exemple suivant affiche des informations sur plusieurs applications associées au AWS compte de l'utilisateur.

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

Sortie :

```
{
    "applicationsInfo": [
        {
            "applicationName": "WordPress_App",
            "applicationId": "d9dd6993-f171-44fa-a811-211e4EXAMPLE",
            "createTime": 1407878168.078,
            "linkedToGitHub": false
        },
        {
            "applicationName": "MyOther_App",
            "applicationId": "8ca57519-31da-42b2-9194-8bb16EXAMPLE",
            "createTime": 1407453571.63,
            "linkedToGitHub": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous BatchGetApplicationsà la section Référence des AWS CLI commandes.

`batch-get-deployment-groups`

L'exemple de code suivant montre comment utiliser`batch-get-deployment-groups`.

AWS CLI

Pour récupérer des informations sur un ou plusieurs groupes de déploiement

L'`batch-get-deployment-groups`exemple suivant extrait des informations sur deux des groupes de déploiement associés à l' CodeDeploy application spécifiée.

```
aws deploy batch-get-deployment-groups \
    --application-name my-codedeploy-application \
    --deployment-group-names "[\"my-deployment-group-1\",\"my-deployment-group-2\"]"
```

Sortie :

```
{
    "deploymentGroupsInfo": [
        {
            "deploymentStyle": {
                "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
                "deploymentType": "IN_PLACE"
            },
            "autoRollbackConfiguration": {
                "enabled": false
            },
            "onPremisesTagSet": {
                "onPremisesTagSetList": []
            },
            "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
            "lastAttemptedDeployment": {
                "endTime": 1556912366.415,
                "status": "Failed",
                "createTime": 1556912355.884,
                "deploymentId": "d-A1B2C3111"
            },
            "autoScalingGroups": [],
```

```
            "deploymentGroupName": "my-deployment-group-1",
            "ec2TagSet": {
                "ec2TagSetList": [
                    [
                        {
                            "Type": "KEY_AND_VALUE",
                            "Value": "my-EC2-instance",
                            "Key": "Name"
                        }
                    ]
                ]
            },
            "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111example",
            "triggerConfigurations": [],
            "applicationName": "my-codedeploy-application",
            "computePlatform": "Server",
            "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
        },
        {
            "deploymentStyle": {
                "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
                "deploymentType": "IN_PLACE"
            },
            "autoRollbackConfiguration": {
                "enabled": false
            },
            "onPremisesTagSet": {
                "onPremisesTagSetList": []
            },
            "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
            "autoScalingGroups": [],
            "deploymentGroupName": "my-deployment-group-2",
            "ec2TagSet": {
                "ec2TagSetList": [
                    [
                        {
                            "Type": "KEY_AND_VALUE",
                            "Value": "my-EC2-instance",
                            "Key": "Name"
                            }
                    ]
                ]
            },
```

```
            "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-22222example",
            "triggerConfigurations": [],
            "applicationName": "my-codedeploy-application",
            "computePlatform": "Server",
            "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
        }
    ],
    "errorMessage": ""
}
```

Pour plus d'informations, consultez BatchGetDeploymentGroups dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous BatchGetDeploymentGroupsà la section Référence des AWS CLI commandes.

`batch-get-deployment-targets`

L'exemple de code suivant montre comment utiliser`batch-get-deployment-targets`.

AWS CLI

Pour récupérer les cibles associées à un déploiement

L'`batch-get-deployment-targets`exemple suivant renvoie des informations sur l'une des cibles associées au déploiement spécifié.

```
aws deploy batch-get-deployment-targets \
    --deployment-id "d-1A2B3C4D5" \
    --target-ids "i-01a2b3c4d5e6f1111"
```

Sortie :

```
{
    "deploymentTargets": [
        {
            "deploymentTargetType": "InstanceTarget",
            "instanceTarget": {
                "lifecycleEvents": [
                    {
                        "startTime": 1556918592.162,
                        "lifecycleEventName": "ApplicationStop",
                        "status": "Succeeded",
```

```
                        "endTime": 1556918592.247,
                        "diagnostics": {
                            "scriptName": "",
                            "errorCode": "Success",
                            "logTail": "",
                            "message": "Succeeded"
                        }
                    },
                    {
                        "startTime": 1556918593.193,
                        "lifecycleEventName": "DownloadBundle",
                        "status": "Succeeded",
                        "endTime": 1556918593.981,
                        "diagnostics": {
                            "scriptName": "",
                            "errorCode": "Success",
                            "logTail": "",
                            "message": "Succeeded"
                        }
                    },
                    {
                        "startTime": 1556918594.805,
                        "lifecycleEventName": "BeforeInstall",
                        "status": "Succeeded",
                        "endTime": 1556918681.807,
                        "diagnostics": {
                            "scriptName": "",
                            "errorCode": "Success",
                            "logTail": "",
                            "message": "Succeeded"
                        }
                    }
                ],
                "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a2b3c4d5e6f1111",
                "deploymentId": "d-1A2B3C4D5",
                "lastUpdatedAt": 1556918687.504,
                "targetId": "i-01a2b3c4d5e6f1111",
                "status": "Succeeded"
            }
        }
    ]
}
```

Pour plus d'informations, consultez BatchGetDeploymentTargets dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous BatchGetDeploymentTargetsà la section Référence des AWS CLI commandes.

`batch-get-deployments`

L'exemple de code suivant montre comment utiliser`batch-get-deployments`.

AWS CLI

Pour obtenir des informations sur les déploiements multiples

L'`batch-get-deployments`exemple suivant affiche des informations sur plusieurs déploiements associés au AWS compte de l'utilisateur.

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

Sortie :

```
{
    "deploymentsInfo": [
        {
            "applicationName": "WordPress_App",
            "status": "Failed",
            "deploymentOverview": {
                "Failed": 0,
                "InProgress": 0,
                "Skipped": 0,
                "Succeeded": 1,
                "Pending": 0
            },
            "deploymentConfigName": "CodeDeployDefault.OneAtATime",
            "creator": "user",
            "deploymentGroupName": "WordPress_DG",
            "revision": {
                "revisionType": "S3",
                "s3Location": {
                "bundleType": "zip",
                "version": "uTecLusEXAMPLEFXtfUcyfV8bEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
```

```
                "key": "WordPressApp.zip"
                }
            },
            "deploymentId": "d-A1B2C3111",
            "createTime": 1408480721.9,
            "completeTime": 1408480741.822
        },
        {
            "applicationName": "MyOther_App",
            "status": "Failed",
            "deploymentOverview": {
                "Failed": 1,
                "InProgress": 0,
                "Skipped": 0,
                "Succeeded": 0,
                "Pending": 0
            },
            "deploymentConfigName": "CodeDeployDefault.OneAtATime",
            "creator": "user",
            "errorInformation": {
                "message": "Deployment failed: Constraint default violated: No hosts
 succeeded.",
                "code": "HEALTH_CONSTRAINTS"
            },
            "deploymentGroupName": "MyOther_DG",
            "revision": {
                "revisionType": "S3",
                "s3Location": {
                "bundleType": "zip",
                "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
                "bucket": "amzn-s3-demo-bucket",
                "key": "MyOtherApp.zip"
                }
            },
            "deploymentId": "d-A1B2C3222",
            "createTime": 1409764576.589,
            "completeTime": 1409764596.101
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous BatchGetDeploymentsà la section Référence des AWS CLI commandes.

batch-get-on-premises-instances

L'exemple de code suivant montre comment utiliserbatch-get-on-premises-instances.

AWS CLI

Pour obtenir des informations sur une ou plusieurs instances locales

L'batch-get-on-premises-instancesexemple suivant permet d'obtenir des informations sur deux instances locales.

```
aws deploy batch-get-on-premises-instances --instance-
names AssetTag12010298EX AssetTag23121309EX
```

Sortie :

```
{
    "instanceInfos": [
        {
            "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag12010298EX",
            "tags": [
                {
                    "Value": "CodeDeployDemo-OnPrem",
                    "Key": "Name"
                }
            ],
            "instanceName": "AssetTag12010298EX",
            "registerTime": 1425579465.228,
            "instanceArn": "arn:aws:codedeploy:us-west-2:123456789012:instance/
AssetTag12010298EX_4IwLNI2Alh"
        },
        {
            "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag23121309EX",
            "tags": [
                {
                    "Value": "CodeDeployDemo-OnPrem",
                    "Key": "Name"
                }
            ],
            "instanceName": "AssetTag23121309EX",
            "registerTime": 1425595585.988,
```

```
            "instanceArn": "arn:aws:codedeploy:us-west-2:80398EXAMPLE:instance/
AssetTag23121309EX_PomUy64Was"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous BatchGetOnPremisesInstancesà la section Référence des AWS CLI commandes.

`continue-deployment`

L'exemple de code suivant montre comment utiliser`continue-deployment`.

AWS CLI

Pour commencer à réacheminer le trafic sans attendre l'expiration d'un délai d'attente spécifié.

L'`continue-deployment`exemple suivant commence à rediriger le trafic depuis les instances de l'environnement d'origine qui sont prêtes à commencer à transférer le trafic vers les instances de l'environnement de remplacement.

```
aws deploy continue-deployment \
    --deployment-id "d-A1B2C3111" \
    --deployment-wait-type "READY_WAIT"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez ContinueDeployment dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous ContinueDeploymentà la section Référence des AWS CLI commandes.

`create-application`

L'exemple de code suivant montre comment utiliser`create-application`.

AWS CLI

Pour créer une application

L'`create-application`exemple suivant crée une application et l'associe au AWS compte de l'utilisateur.

```
aws deploy create-application --application-name MyOther_App
```

Sortie :

```
{
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateApplicationà la section Référence des AWS CLI commandes.

`create-deployment-config`

L'exemple de code suivant montre comment utiliser`create-deployment-config`.

AWS CLI

Pour créer une configuration de déploiement personnalisée

L'`create-deployment-config`exemple suivant crée une configuration de déploiement personnalisée et l'associe au AWS compte de l'utilisateur.

```
aws deploy create-deployment-config \
    --deployment-config-name ThreeQuartersHealthy \
    --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

Sortie :

```
{
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentConfigà la section Référence des AWS CLI commandes.

`create-deployment-group`

L'exemple de code suivant montre comment utiliser`create-deployment-group`.

AWS CLI

Pour créer un groupe de déploiement

L'`create-deployment-group`exemple suivant crée un groupe de déploiement et l'associe à l'application spécifiée et au AWS compte de l'utilisateur.

```
aws deploy create-deployment-group \
    --application-name WordPress_App \
    --auto-scaling-groups CodeDeployDemo-ASG \
    --deployment-config-name CodeDeployDefault.OneAtATime \
    --deployment-group-name WordPress_DG \
    --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \
    --service-role-arn arn:aws:iam::123456789012:role/CodeDeployDemoRole
```

Sortie :

```
{
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentGroupà la section Référence des AWS CLI commandes.

`create-deployment`

L'exemple de code suivant montre comment utiliser`create-deployment`.

AWS CLI

Exemple 1 : pour créer un CodeDeploy déploiement à l'aide de la plateforme de EC2 calcul /on-premises

L'`create-deployment`exemple suivant crée un déploiement et l'associe au AWS compte de l'utilisateur.

```
aws deploy create-deployment \
```

```
    --application-name WordPress_App \
    --deployment-config-name CodeDeployDefault.OneAtATime \
    --deployment-group-name WordPress_DG \
    --description "My demo deployment" \
    --s3-location bucket=amzn-s3-demo-
bucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,key=WordPressApp.zip
```

Sortie :

```
{
    "deploymentId": "d-A1B2C3111"
}
```

Exemple 2 : pour créer un CodeDeploy déploiement à l'aide de la plateforme de calcul Amazon ECS

L'`create-deployment`exemple suivant utilise les deux fichiers suivants pour déployer un service Amazon ECS.

Contenu du `create-deployment.json` fichier :

```
{
    "applicationName": "ecs-deployment",
    "deploymentGroupName": "ecs-deployment-dg",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bucket": "ecs-deployment-bucket",
            "key": "appspec.yaml",
            "bundleType": "YAML"
        }
    }
}
```

Ce fichier extrait à son tour le fichier `appspec.yaml` suivant dans un compartiment S3 appelé`ecs-deployment-bucket`.

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
```

```
        TaskDefinition: "arn:aws:ecs:region:123456789012:task-definition/ecs-task-
def:2"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80
        PlatformVersion: "LATEST"
```

Commande :

```
aws deploy create-deployment \
    --cli-input-json file://create-deployment.json \
    --region us-east-1
```

Sortie :

```
{
    "deploymentId": "d-1234ABCDE"
}
```

Pour plus d'informations, consultez CreateDeployment dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentà la section Référence des AWS CLI commandes.

`delete-application`

L'exemple de code suivant montre comment utiliser`delete-application`.

AWS CLI

Pour supprimer une application

L'`delete-application`exemple suivant supprime l'application spécifiée associée au AWS compte de l'utilisateur.

```
aws deploy delete-application --application-name WordPress_App
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteApplicationà la section Référence des AWS CLI commandes.

`delete-deployment-config`

L'exemple de code suivant montre comment utiliser`delete-deployment-config`.

AWS CLI

Pour supprimer une configuration de déploiement

L'`delete-deployment-config`exemple suivant supprime une configuration de déploiement personnalisée associée au AWS compte de l'utilisateur.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteDeploymentConfigà la section Référence des AWS CLI commandes.

`delete-deployment-group`

L'exemple de code suivant montre comment utiliser`delete-deployment-group`.

AWS CLI

Pour supprimer un groupe de déploiement

L'`delete-deployment-group`exemple suivant supprime un groupe de déploiement associé à l'application spécifiée.

```
aws deploy delete-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

Sortie :

```
{
    "hooksNotCleanedUp": []
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDeploymentGroupà la section Référence des AWS CLI commandes.

`delete-git-hub-account-token`

L'exemple de code suivant montre comment utiliser`delete-git-hub-account-token`.

AWS CLI

Pour supprimer une connexion à un GitHub compte

L'`delete-git-hub-account-token`exemple suivant supprime la connexion du GitHub compte spécifié.

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

Sortie :

```
{
    "tokenName": "my-github-account"
}
```

Pour plus d'informations, consultez DeleteGitHubAccountToken dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous DeleteGitHubAccountTokenà la section Référence des AWS CLI commandes.

`deregister-on-premises-instance`

L'exemple de code suivant montre comment utiliser`deregister-on-premises-instance`.

AWS CLI

Pour annuler l'enregistrement d'une instance locale

L'`deregister-on-premises-instance`exemple suivant annule l'enregistrement d'une instance locale auprès de l'instance AWS CodeDeploy, mais il ne supprime pas l'utilisateur IAM associé à l'instance, et il ne dissocie pas les balises de l'instance sur site de AWS CodeDeploy l'instance. Il ne désinstalle pas non plus l' AWS CodeDeploy agent de l'instance ni ne supprime le fichier de configuration local de l'instance.

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeregisterOnPremisesInstanceà la section Référence des AWS CLI commandes.

`deregister`

L'exemple de code suivant montre comment utiliser`deregister`.

AWS CLI

Pour annuler l'enregistrement d'une instance locale

L'`deregister`exemple suivant annule l'enregistrement d'une instance locale avec. AWS CodeDeploy Il ne supprime pas l'utilisateur IAM associé à l'instance. Il dissocie AWS CodeDeploy les balises locales de l'instance. Il ne désinstalle pas l' AWS CodeDeploy agent de l'instance et ne supprime pas le fichier de configuration local de l'instance.

```
aws deploy deregister \
    --instance-name AssetTag12010298EX \
    --no-delete-iam-user \
    --region us-west-2
```

Sortie :

```
Retrieving on-premises instance information... DONE
IamUserArn: arn:aws:iam::80398EXAMPLE:user/AWS/CodeDeploy/AssetTag12010298EX
Tags: Key=Name,Value=CodeDeployDemo-OnPrem
Removing tags from the on-premises instance... DONE
Deregistering the on-premises instance... DONE
Run the following command on the on-premises instance to uninstall the codedeploy-
agent:
aws deploy uninstall
```

- Pour plus de détails sur l'API, voir Désenregistrer dans AWS CLI Command Reference.

`get-application-revision`

L'exemple de code suivant montre comment utiliser`get-application-revision`.

AWS CLI

Pour obtenir des informations sur la révision d'une application

L'`get-application-revision`exemple suivant affiche des informations sur une révision d'application associée à l'application spécifiée.

```
aws deploy get-application-revision \
    --application-name WordPress_App \
    --s3-location bucket=amzn-s3-demo-
bucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,key=WordPressApp.zip
```

Sortie :

```
{
    "applicationName": "WordPress_App",
    "revisionInfo": {
        "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
        "registerTime": 1411076520.009,
        "deploymentGroups": "WordPress_DG",
        "lastUsedTime": 1411076520.009,
        "firstUsedTime": 1411076520.009
    },
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bundleType": "zip",
            "eTag": "dd56cfdEXAMPLE8e768f9d77fEXAMPLE",
            "bucket": "amzn-s3-demo-bucket",
            "key": "WordPressApp.zip"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetApplicationRevisionà la section Référence des AWS CLI commandes.

`get-application`

L'exemple de code suivant montre comment utiliser`get-application`.

AWS CLI

Pour obtenir des informations sur une application

L'`get-application`exemple suivant affiche des informations sur une application associée au AWS compte de l'utilisateur.

```
aws deploy get-application --application-name WordPress_App
```

Sortie :

```
{
    "application": {
        "applicationName": "WordPress_App",
        "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "createTime": 1407878168.078,
        "linkedToGitHub": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetApplicationà la section Référence des AWS CLI commandes.

`get-deployment-config`

L'exemple de code suivant montre comment utiliser`get-deployment-config`.

AWS CLI

Pour obtenir des informations sur une configuration de déploiement

L'`get-deployment-config`exemple suivant affiche des informations sur une configuration de déploiement associée au AWS compte de l'utilisateur.

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Sortie :

```
{
    "deploymentConfigInfo": {
        "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "minimumHealthyHosts": {
            "type": "FLEET_PERCENT",
            "value": 75
```

```
        },
        "createTime": 1411081164.379,
        "deploymentConfigName": "ThreeQuartersHealthy"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentConfigà la section Référence des AWS CLI commandes.

`get-deployment-group`

L'exemple de code suivant montre comment utiliser`get-deployment-group`.

AWS CLI

Pour afficher les informations relatives à un groupe de déploiement

L'`get-deployment-group`exemple suivant affiche des informations sur un groupe de déploiement associé à l'application spécifiée.

```
aws deploy get-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

Sortie :

```
{
    "deploymentGroupInfo": {
        "applicationName": "WordPress_App",
        "autoScalingGroups": [
            "CodeDeployDemo-ASG"
        ],
        "deploymentConfigName": "CodeDeployDefault.OneAtATime",
        "ec2TagFilters": [
            {
                "Type": "KEY_AND_VALUE",
                "Value": "CodeDeployDemo",
                "Key": "Name"
            }
        ],
        "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "serviceRoleArn": "arn:aws:iam::123456789012:role/CodeDeployDemoRole",
```

```
        "deploymentGroupName": "WordPress_DG"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentGroupà la section Référence des AWS CLI commandes.

`get-deployment-instance`

L'exemple de code suivant montre comment utiliser`get-deployment-instance`.

AWS CLI

Pour obtenir des informations sur une instance de déploiement

L'`get-deployment-instance`exemple suivant affiche des informations sur une instance de déploiement associée au déploiement spécifié.

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-
id i-902e9fEX
```

Sortie :

```
{
    "instanceSummary": {
        "instanceId": "arn:aws:ec2:us-east-1:80398EXAMPLE:instance/i-902e9fEX",
        "lifecycleEvents": [
            {
                "status": "Succeeded",
                "endTime": 1408480726.569,
                "startTime": 1408480726.437,
                "lifecycleEventName": "ApplicationStop"
            },
            {
                "status": "Succeeded",
                "endTime": 1408480728.016,
                "startTime": 1408480727.665,
                "lifecycleEventName": "DownloadBundle"
            },
            {
                "status": "Succeeded",
```

```
                "endTime": 1408480729.744,
                "startTime": 1408480729.125,
                "lifecycleEventName": "BeforeInstall"
            },
            {
                "status": "Succeeded",
                "endTime": 1408480730.979,
                "startTime": 1408480730.844,
                "lifecycleEventName": "Install"
            },
            {
                "status": "Failed",
                "endTime": 1408480732.603,
                "startTime": 1408480732.1,
                "lifecycleEventName": "AfterInstall"
            },
            {
                "status": "Skipped",
                "endTime": 1408480732.606,
                "lifecycleEventName": "ApplicationStart"
            },
            {
                "status": "Skipped",
                "endTime": 1408480732.606,
                "lifecycleEventName": "ValidateService"
            }
        ],
        "deploymentId": "d-QA4G4F9EX",
        "lastUpdatedAt": 1408480733.152,
        "status": "Failed"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentInstanceà la section Référence des AWS CLI commandes.

`get-deployment-target`

L'exemple de code suivant montre comment utiliser`get-deployment-target`.

AWS CLI

Pour renvoyer des informations sur une cible de déploiement

L'`get-deployment-target`exemple suivant renvoie des informations sur une cible de déploiement associée au déploiement spécifié.

```
aws deploy get-deployment-target \
    --deployment-id "d-A1B2C3111" \
    --target-id "i-a1b2c3d4e5f611111"
```

Sortie :

```
{
    "deploymentTarget": {
        "deploymentTargetType": "InstanceTarget",
        "instanceTarget": {
            "lastUpdatedAt": 1556918687.504,
            "targetId": "i-a1b2c3d4e5f611111",
            "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-
a1b2c3d4e5f611111",
            "status": "Succeeded",
            "lifecycleEvents": [
                {
                    "status": "Succeeded",
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
                        "scriptName": ""
                    },
                    "lifecycleEventName": "ApplicationStop",
                    "startTime": 1556918592.162,
                    "endTime": 1556918592.247
                },
                {
                    "status": "Succeeded",
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
                        "scriptName": ""
                    },
                    "lifecycleEventName": "DownloadBundle",
                    "startTime": 1556918593.193,
                    "endTime": 1556918593.981
                },
```

```
            {
                "status": "Succeeded",
                "diagnostics": {
                    "errorCode": "Success",
                    "message": "Succeeded",
                    "logTail": "",
                    "scriptName": ""
                },
                "lifecycleEventName": "BeforeInstall",
                "startTime": 1556918594.805,
                "endTime": 1556918681.807
            },
            {
                "status": "Succeeded",
                "diagnostics": {
                    "errorCode": "Success",
                    "message": "Succeeded",
                    "logTail": "",
                    "scriptName": ""
                },
                "lifecycleEventName": "Install",
                "startTime": 1556918682.696,
                "endTime": 1556918683.005
            },
            {
                "status": "Succeeded",
                "diagnostics": {
                    "errorCode": "Success",
                    "message": "Succeeded",
                    "logTail": "",
                    "scriptName": ""
                },
                "lifecycleEventName": "AfterInstall",
                "startTime": 1556918684.135,
                "endTime": 1556918684.216
            },
            {
                "status": "Succeeded",
                "diagnostics": {
                    "errorCode": "Success",
                    "message": "Succeeded",
                    "logTail": "",
                    "scriptName": ""
                },
```

```
                    "lifecycleEventName": "ApplicationStart",
                    "startTime": 1556918685.211,
                    "endTime": 1556918685.295
                },
                {
                    "status": "Succeeded",
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
                        "scriptName": ""
                    },
                    "lifecycleEventName": "ValidateService",
                    "startTime": 1556918686.65,
                    "endTime": 1556918686.747
                }
            ],
            "deploymentId": "d-A1B2C3111"
        }
    }
}
```

Pour plus d'informations, consultez GetDeploymentTarget dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous GetDeploymentTargetà la section Référence des AWS CLI commandes.

`get-deployment`

L'exemple de code suivant montre comment utiliser`get-deployment`.

AWS CLI

Pour obtenir des informations sur un déploiement

L'`get-deployment`exemple suivant affiche des informations sur un déploiement associé au AWS compte de l'utilisateur.

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

Sortie :

```
{
    "deploymentInfo": {
        "applicationName": "WordPress_App",
        "status": "Succeeded",
        "deploymentOverview": {
            "Failed": 0,
            "InProgress": 0,
            "Skipped": 0,
            "Succeeded": 1,
            "Pending": 0
        },
        "deploymentConfigName": "CodeDeployDefault.OneAtATime",
        "creator": "user",
        "description": "My WordPress app deployment",
        "revision": {
            "revisionType": "S3",
            "s3Location":  {
            "bundleType": "zip",
            "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
            "bucket": "amzn-s3-demo-bucket",
            "key": "WordPressApp.zip"
            }
        },
        "deploymentId": "d-A1B2C3123",
        "deploymentGroupName": "WordPress_DG",
        "createTime": 1409764576.589,
        "completeTime": 1409764596.101,
        "ignoreApplicationStopFailures": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentà la section Référence des AWS CLI commandes.

`get-on-premises-instance`

L'exemple de code suivant montre comment utiliser`get-on-premises-instance`.

AWS CLI

Pour obtenir des informations sur une instance locale

L'`get-on-premises-instance`exemple suivant extrait des informations sur l'instance locale spécifiée.

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Sortie :

```
{
    "instanceInfo": {
    "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag12010298EX",
        "tags": [
        {
            "Value": "CodeDeployDemo-OnPrem",
            "Key": "Name"
        }
        ],
        "instanceName": "AssetTag12010298EX",
        "registerTime": 1425579465.228,
        "instanceArn": "arn:aws:codedeploy:us-east-1:123456789012:instance/
AssetTag12010298EX_4IwLNI2Alh"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetOnPremisesInstanceà la section Référence des AWS CLI commandes.

`install`

L'exemple de code suivant montre comment utiliser`install`.

AWS CLI

Pour installer une instance sur site

L'`install`exemple suivant copie le fichier de configuration sur site de l'emplacement spécifié sur l'instance vers l'emplacement sur l'instance où l' AWS CodeDeploy agent s'attend à le trouver. Il installe également l' AWS CodeDeploy agent sur l'instance. Il ne crée aucun utilisateur IAM, n'enregistre pas l'instance locale avec AWS CodeDeploy elle et n'associe aucune balise d'instance locale à AWS CodeDeploy l'instance.

```
aws deploy install \
    --override-config \
    --config-file C:\temp\codedeploy.onpremises.yml \
    --region us-west-2 \
    --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-agent.msi
```

Sortie :

```
Creating the on-premises instance configuration file... DONE
Installing the AWS CodeDeploy Agent... DONE
```

- Pour plus de détails sur l'API, voir Installer dans le manuel de référence des AWS CLI commandes.

`list-application-revisions`

L'exemple de code suivant montre comment utiliser`list-application-revisions`.

AWS CLI

Pour obtenir des informations sur les révisions des applications

L'`list-application-revisions`exemple suivant affiche des informations sur toutes les révisions d'applications associées à l'application spécifiée.

```
aws deploy list-application-revisions \
    --application-name WordPress_App \
    --s-3-bucket amzn-s3-demo-bucket \
    --deployed exclude \
    --s-3-key-prefix WordPress_ \
    --sort-by lastUsedTime \
    --sort-order descending
```

Sortie :

```
{
    "revisions": [
        {
            "revisionType": "S3",
            "s3Location": {
```

```
                "version": "uTecLusvCB_JqHFXtfUcyfV8bEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
                "key": "WordPress_App.zip",
                "bundleType": "zip"
            }
        },
        {
            "revisionType": "S3",
            "s3Location": {
                "version": "tMk.UxgDpMEVb7V187ZM6wVAWEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
                "key": "WordPress_App_2-0.zip",
                "bundleType": "zip"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListApplicationRevisionsà la section Référence des AWS CLI commandes.

`list-applications`

L'exemple de code suivant montre comment utiliser`list-applications`.

AWS CLI

Pour obtenir des informations sur les applications

L'`list-applications`exemple suivant affiche des informations sur toutes les applications associées au AWS compte de l'utilisateur.

```
aws deploy list-applications
```

Sortie :

```
{
    "applications": [
        "WordPress_App",
        "MyOther_App"
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListApplicationsà la section Référence des AWS CLI commandes.

`list-deployment-configs`

L'exemple de code suivant montre comment utiliser`list-deployment-configs`.

AWS CLI

Pour obtenir des informations sur les configurations de déploiement

L'`list-deployment-configs`exemple suivant affiche des informations sur toutes les configurations de déploiement associées au AWS compte de l'utilisateur.

```
aws deploy list-deployment-configs
```

Sortie :

```
{
    "deploymentConfigsList": [
        "ThreeQuartersHealthy",
        "CodeDeployDefault.AllAtOnce",
        "CodeDeployDefault.HalfAtATime",
        "CodeDeployDefault.OneAtATime"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeploymentConfigsà la section Référence des AWS CLI commandes.

`list-deployment-groups`

L'exemple de code suivant montre comment utiliser`list-deployment-groups`.

AWS CLI

Pour obtenir des informations sur les groupes de déploiement

L'`list-deployment-groups`exemple suivant affiche des informations sur tous les groupes de déploiement associés à l'application spécifiée.

```
aws deploy list-deployment-groups --application-name WordPress_App
```

Sortie :

```
{
    "applicationName": "WordPress_App",
    "deploymentGroups": [
        "WordPress_DG",
        "WordPress_Beta_DG"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeploymentGroupsà la section Référence des AWS CLI commandes.

`list-deployment-instances`

L'exemple de code suivant montre comment utiliser`list-deployment-instances`.

AWS CLI

Pour obtenir des informations sur les instances de déploiement

L'`list-deployment-instances`exemple suivant affiche des informations sur toutes les instances de déploiement associées au déploiement spécifié.

```
aws deploy list-deployment-instances \
    --deployment-id d-A1B2C3111 \
    --instance-status-filter Succeeded
```

Sortie :

```
{
    "instancesList": [
        "i-EXAMPLE11",
        "i-EXAMPLE22"
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeploymentInstancesà la section Référence des AWS CLI commandes.

`list-deployment-targets`

L'exemple de code suivant montre comment utiliser`list-deployment-targets`.

AWS CLI

Pour récupérer la liste IDs des cibles associées à un déploiement

L'`list-deployment-targets`exemple suivant extrait une liste de cibles IDs associées à des déploiements dont le statut est « Échec » ou « ». InProgress

```
aws deploy list-deployment-targets \
    --deployment-id "d-A1B2C3111" \
    --target-filters "{\"TargetStatus\":[\"Failed\",\"InProgress\"]}"
```

Sortie :

```
{
    "targetIds": [
        "i-0f1558aaf90e5f1f9"
    ]
}
```

Pour plus d'informations, consultez ListDeploymentTargets dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous ListDeploymentTargetsà la section Référence des AWS CLI commandes.

`list-deployments`

L'exemple de code suivant montre comment utiliser`list-deployments`.

AWS CLI

Pour obtenir des informations sur les déploiements

L'`list-deployments`exemple suivant affiche des informations sur tous les déploiements associés à l'application et au groupe de déploiement spécifiés.

```
aws deploy list-deployments \
    --application-name WordPress_App \
    --create-time-range start=2014-08-19T00:00:00,end=2014-08-20T00:00:00 \
    --deployment-group-name WordPress_DG \
    --include-only-statuses Failed
```

Sortie :

```
{
    "deployments": [
        "d-EXAMPLE11",
        "d-EXAMPLE22",
        "d-EXAMPLE33"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeploymentsà la section Référence des AWS CLI commandes.

`list-git-hub-account-token-names`

L'exemple de code suivant montre comment utiliser`list-git-hub-account-token-names`.

AWS CLI

Pour répertorie les noms des connexions enregistrées aux GitHub comptes

L'`list-git-hub-account-token-names`exemple suivant répertorie les noms des connexions enregistrées aux GitHub comptes de l' AWS utilisateur actuel.

```
aws deploy list-git-hub-account-token-names
```

Sortie :

```
{
    "tokenNameList": [
        "my-first-token",
        "my-second-token",
```

```
        "my-third-token"
    ]
}
```

Pour plus d'informations, consultez ListGitHubAccountTokenNames dans la Référence d'API AWS CodeDeploy .

- Pour plus de détails sur l'API, reportez-vous ListGitHubAccountTokenNamesà la section Référence des AWS CLI commandes.

`list-on-premises-instances`

L'exemple de code suivant montre comment utiliser`list-on-premises-instances`.

AWS CLI

Pour obtenir des informations sur une ou plusieurs instances locales

L'`list-on-premises-instances`exemple suivant extrait une liste des noms d'instances locales disponibles pour les instances enregistrées dans l'instance AWS CodeDeploy et auxquelles est également associée la balise d'instance locale spécifiée. AWS CodeDeploy

```
aws deploy list-on-premises-instances \
    --registration-status Registered \
    --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

Sortie :

```
{
    "instanceNames": [
        "AssetTag12010298EX"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListOnPremisesInstancesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource (application)

L'`list-tags-for-resource`exemple suivant répertorie les balises appliquées à une application nommée TestApp in CodeDeploy.

```
aws deploy list-tags-for-resource \
    --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Type",
            "Value": "testType"
        },
        {
            "Key": "Name",
            "Value": "testName"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des instances pour les groupes de déploiement CodeDeploy dans le Guide de l'AWS CodeDeploy utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`push`

L'exemple de code suivant montre comment utiliser`push`.

AWS CLI

Pour regrouper et déployer une révision d'application AWS CodeDeploy compatible sur Amazon S3

L'`push`exemple suivant regroupe et déploie une révision d'application sur Amazon S3, puis associe la révision de l'application à l'application spécifiée.

```
aws deploy push \
    --application-name WordPress_App \
    --description "This is my deployment" \
    --ignore-hidden-files \
    --s3-location s3://amzn-s3-demo-bucket/WordPressApp.zip \
    --source /tmp/MyLocalDeploymentFolder/
```

Le résultat décrit comment utiliser la `create-deployment` commande pour créer un déploiement qui utilise la révision de l'application téléchargée.

```
To deploy with this revision, run:
aws deploy create-deployment --application-name WordPress_App --
deployment-config-name <deployment-config-name> --deployment-group-
name <deployment-group-name> --s3-location bucket=amzn-s3-demo-
bucket,key=WordPressApp.zip,bundleType=zip,eTag="cecc9b8EXAMPLE50a6e71fdb88EXAMPLE",version=
```

- Pour plus de détails sur l'API, reportez-vous à la section Push in AWS CLI Command Reference.

`register-application-revision`

L'exemple de code suivant montre comment utiliser`register-application-revision`.

AWS CLI

Pour enregistrer les informations relatives à une révision d'application déjà téléchargée

L'`register-application-revision`exemple suivant enregistre des informations sur une révision d'application déjà chargée et stockée dans Amazon S3 avec. AWS CodeDeploy

```
aws deploy register-application-revision \
    --application-name WordPress_App \
    --description "Revised WordPress application" \
    --s3-location bucket=amzn-s3-demo-
bucket,key=RevisedWordPressApp.zip,bundleType=zip,eTag=cecc9b8a08eac650a6e71fdb88EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RegisterApplicationRevisionà la section Référence des AWS CLI commandes.

`register-on-premises-instance`

L'exemple de code suivant montre comment utiliser`register-on-premises-instance`.

AWS CLI

Pour enregistrer une instance sur site

L'`register-on-premises-instance`exemple suivant enregistre une instance locale auprès AWS CodeDeploy de. Il ne crée pas l'utilisateur IAM spécifié et n'associe dans AWS CodeDeploy aucune instance locale des balises à l'instance enregistrée.

```
aws deploy register-on-premises-instance \
    --instance-name AssetTag12010298EX \
    --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RegisterOnPremisesInstanceà la section Référence des AWS CLI commandes.

`register`

L'exemple de code suivant montre comment utiliser`register`.

AWS CLI

Pour enregistrer une instance sur site

L'`register`exemple suivant enregistre une instance locale auprès de l'instance enregistrée AWS CodeDeploy, AWS CodeDeploy l'associe à la balise d'instance locale spécifiée et crée un fichier de configuration local qui peut être copié sur l'instance. Il ne crée pas l'utilisateur IAM et n'installe pas l' AWS CodeDeploy agent sur l'instance.

```
aws deploy register \
    --instance-name AssetTag12010298EX \
    --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployUser-OnPrem \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem \
    --region us-west-2
```

Sortie :

```
Registering the on-premises instance... DONE
Adding tags to the on-premises instance... DONE
Copy the on-premises configuration file named codedeploy.onpremises.yml to the on-
premises instance, and run the following command on the on-premises instance to
 install and configure the AWS CodeDeploy Agent:
aws deploy install --config-file codedeploy.onpremises.yml
```

- Pour plus de détails sur l'API, voir Register in AWS CLI Command Reference.

`remove-tags-from-on-premises-instances`

L'exemple de code suivant montre comment utiliser`remove-tags-from-on-premises-instances`.

AWS CLI

Pour supprimer des balises d'une ou de plusieurs instances locales

L'`remove-tags-from-on-premises-instances`exemple suivant dissocie les balises locales spécifiées AWS CodeDeploy des instances locales. Il ne désenregistre pas les instances locales dans l'instance AWS CodeDeploy, ne désinstalle pas l' AWS CodeDeploy agent de l'instance, ne supprime pas le fichier de configuration sur site des instances, ni ne supprime les utilisateurs IAM associés aux instances.

```
aws deploy remove-tags-from-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RemoveTagsFromOnPremisesInstancesà la section Référence des AWS CLI commandes.

`stop-deployment`

L'exemple de code suivant montre comment utiliser`stop-deployment`.

AWS CLI

Pour tenter d'arrêter un déploiement

L'`stop-deployment`exemple suivant tente d'arrêter un déploiement en cours associé au AWS compte de l'utilisateur.

aws deploy stop-deployment --deployment-id d-a1b2c3111

Sortie :

```
{
    "status": "Succeeded",
    "statusMessage": "No more commands will be scheduled for execution in the
 deployment instances"
}
```

- Pour plus de détails sur l'API, reportez-vous StopDeploymentà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource (application)

L'`tag-resource`exemple suivant ajoute deux balises avec les clés Name et Type, et les valeurs TestName et TestType à une application nommée TestApp in. CodeDeploy :

```
aws deploy tag-resource \
    --resource-arn  arn:aws:codedeploy:us-west-2:111122223333:application:testApp \
    --tags Key=Name,Value=testName Key=Type,Value=testType
```

En cas de succès, cette commande ne produit aucune sortie.

Pour plus d'informations, consultez la section Marquage des instances pour les groupes de déploiement CodeDeploy dans le Guide de l'AWS CodeDeploy utilisateur.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`uninstall`

L'exemple de code suivant montre comment utiliser`uninstall`.

AWS CLI

Pour désinstaller une instance locale

L'`uninstall`exemple suivant désinstalle l' AWS CodeDeploy agent de l'instance locale et supprime le fichier de configuration local de l'instance. Il ne désenregistre pas l'instance dans l'instance AWS CodeDeploy, ne dissocie aucune balise d'instance locale AWS CodeDeploy de l'instance, ni ne supprime l'utilisateur IAM associé à l'instance.

```
aws deploy uninstall
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, voir Désinstaller dans le manuel de référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource (application)

L'`untag-resource`exemple suivant supprime deux balises avec les clés Name et Type d'une application nommée TestApp in CodeDeploy.

```
aws deploy untag-resource \
    --resource-arn  arn:aws:codedeploy:us-west-2:111122223333:application:testApp \
    --tag-keys Name Type
```

En cas de succès, cette commande ne produit aucune sortie.

Pour plus d'informations, consultez la section Marquage des instances pour les groupes de déploiement CodeDeploy dans le Guide de l'AWS CodeDeploy utilisateur.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-application`

L'exemple de code suivant montre comment utiliser`update-application`.

AWS CLI

Pour modifier les détails d'une demande

L'`update-application`exemple suivant modifie le nom d'une application associée au AWS compte de l'utilisateur.

```
aws deploy update-application \
    --application-name WordPress_App \
    --new-application-name My_WordPress_App
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateApplicationà la section Référence des AWS CLI commandes.

`update-deployment-group`

L'exemple de code suivant montre comment utiliser`update-deployment-group`.

AWS CLI

Pour modifier les informations relatives à un groupe de déploiement

L'`update-deployment-group`exemple suivant modifie les paramètres d'un groupe de déploiement associé à l'application spécifiée.

```
aws deploy update-deployment-group \
    --application-name WordPress_App \
    --auto-scaling-groups My_CodeDeployDemo_ASG \
    --current-deployment-group-name WordPress_DG \
    --deployment-config-name CodeDeployDefault.AllAtOnce \
    --ec2-tag-filters Key=Name,Type=KEY_AND_VALUE,Value=My_CodeDeployDemo \
    --new-deployment-group-name My_WordPress_DepGroup \
    --service-role-arn arn:aws:iam::80398EXAMPLE:role/CodeDeployDemo-2
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateDeploymentGroupà la section Référence des AWS CLI commandes.

CodeGuru Exemples de réviseurs utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with CodeGuru Reviewer.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-repository`

L'exemple de code suivant montre comment utiliser`associate-repository`.

AWS CLI

Exemple 1 : pour créer une association de référentiels Bitbucket

L'`associate-repository`exemple suivant crée une association de dépôt à l'aide d'un dépôt Bitbucket existant.

```
aws codeguru-reviewer associate-repository \
    --repository 'Bitbucket={Owner=sample-owner, Name=mySampleRepo,
 ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

Sortie :

```
{
    "RepositoryAssociation": {
        "ProviderType": "Bitbucket",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596216896.979,
```

```
        "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "State": "Associating",
        "StateReason": "Pending Repository Association",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "Owner": "sample-owner"
    }
}
```

Pour plus d'informations, consultez la section Créer une association de référentiel Bitbucket dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

Exemple 2 : pour créer une association de référentiels GitHub d'entreprise

L'`associate-repository`exemple suivant crée une association de référentiel à l'aide d'un référentiel GitHub d'entreprise existant.

```
aws codeguru-reviewer associate-repository \
    --repository 'GitHubEnterpriseServer={Owner=sample-owner, Name=mySampleRepo,
 ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

Sortie :

```
{
    "RepositoryAssociation": {
        "ProviderType": "GitHubEnterpriseServer",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596216896.979,
        "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "State": "Associating",
        "StateReason": "Pending Repository Association",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "Owner": "sample-owner"
    }
```

```
}
```

Pour plus d'informations, consultez la section Créer une association de référentiels de serveurs GitHub d'entreprise dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur d'Amazon Codeguru Reviewer.

Exemple 3 : pour créer une association de AWS CodeCommit référentiels

L'`associate-repository`exemple suivant crée une association de référentiel à l'aide d'un AWS CodeCommit référentiel existant.

```
aws codeguru-reviewer associate-repository \
    --repository CodeCommit={Name=mySampleRepo}
```

Sortie :

```
{
    "RepositoryAssociation": {
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "My-ecs-beta-repo",
        "LastUpdatedTimeStamp": 1595634764.029,
        "ProviderType": "CodeCommit",
        "CreatedTimeStamp": 1595634764.029,
        "Owner": "544120495673",
        "State": "Associating",
        "StateReason": "Pending Repository Association",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:544120495673:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez la section Créer une association de AWS CodeCommit référentiels dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous AssociateRepositoryà la section Référence des AWS CLI commandes.

`create-code-review`

L'exemple de code suivant montre comment utiliser`create-code-review`.

AWS CLI

Pour créer une révision de code.

Ce qui suit `create-code-review` crée une révision du code dans la `mainline` branche d'un AWS CodeCommit référentiel nommé`my-repository-name`.

```
aws codeguru-reviewer create-code-review \
    --name my-code-review \
    --repository-association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --type '{"RepositoryAnalysis": {"RepositoryHead": {"BranchName": "mainline"}}}'
```

Sortie :

```
{
    "CodeReview": {
        "Name": "my-code-review",
        "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-code-review",
        "RepositoryName": "my-repository-name",
        "Owner": "123456789012",
        "ProviderType": "CodeCommit",
        "State": "Pending",
        "StateReason": "CodeGuru Reviewer has received the request, and a code
 review is scheduled.",
        "CreatedTimeStamp": 1618873489.195,
        "LastUpdatedTimeStamp": 1618873489.195,
        "Type": "RepositoryAnalysis",
        "SourceCodeType": {
            "RepositoryHead": {
                "BranchName": "mainline"
            }
        },
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez la section Créer des révisions de code dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous CreateCodeReviewà la section Référence des AWS CLI commandes.

`describe-code-review`

L'exemple de code suivant montre comment utiliser`describe-code-review`.

AWS CLI

Répertoriez les détails d'une révision de code.

Vous trouverez `describe-code-review` ci-dessous des informations sur la révision du code dans la branche « principale » d'un AWS CodeCommit référentiel nommé « my-repo-name ».

```
aws codeguru-reviewer put-recommendation-feedback \
    --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
    --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \
    --reactions ThumbsUp
```

Sortie

```
{
        "CodeReview": {
            "Name": "My-ecs-beta-repo-master-xs6di4kfd4j269dz",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-repo-name",
            "RepositoryName": "My-ecs-beta-repo",
            "Owner": "123456789012",
            "ProviderType": "CodeCommit",
            "State": "Pending",
            "StateReason": "CodeGuru Reviewer is reviewing the source code.",
            "CreatedTimeStamp": 1618874226.226,
            "LastUpdatedTimeStamp": 1618874233.689,
            "Type": "RepositoryAnalysis",
            "SourceCodeType": {
                "RepositoryHead": {
                    "BranchName": "mainline"
                }
```

```
            },
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    }
```

Pour plus d'informations, consultez la section Afficher les détails de la révision du code dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous DescribeCodeReviewà la section Référence des AWS CLI commandes.

`describe-recommendation-feedback`

L'exemple de code suivant montre comment utiliser`describe-recommendation-feedback`.

AWS CLI

Pour consulter les informations relatives aux commentaires sur une recommandation

Ce qui suit `describe-recommendation-feedback` affiche des informations sur les commentaires relatifs à une recommandation. Cette recommandation entraîne une ThumbsUp réaction.

```
aws codeguru-reviewer describe-recommendation-feedback \
    --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
    --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb
```

Sortie :

```
{
    "RecommendationFeedback": {
        "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678",
        "RecommendationId":
 "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
        "Reactions": [
            "ThumbsUp"
```

```
        ],
        "UserId": "aws-user-id",
        "CreatedTimeStamp": 1618877070.313,
        "LastUpdatedTimeStamp": 1618877948.881
    }
}
```

Pour plus d'informations, consultez les sections Afficher les recommandations et envoyer des commentaires et Étape 4 : Fournir des commentaires dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous DescribeRecommendationFeedbackà la section Référence des AWS CLI commandes.

`describe-repository-association`

L'exemple de code suivant montre comment utiliser`describe-repository-association`.

AWS CLI

Exemple 1 : pour renvoyer des informations sur une association de GitHub référentiels

L'`describe-repository-association`exemple suivant renvoie des informations sur une association de référentiel qui utilise un référentiel GitHub d'entreprise et qui est dans l'`Associated`état.

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RepositoryAssociation": {
        "AssociationId": "b822717e-0711-4e8a-bada-0e738289c75e",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1588102637.649,
        "ProviderType": "GitHub",
        "CreatedTimeStamp": 1588102615.636,
        "Owner": "sample-owner",
        "State": "Associated",
        "StateReason": "Pull Request Notification configuration successful",
```

```
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez la section Créer une association de référentiels de serveurs d' GitHub entreprise dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

Exemple 2 : pour renvoyer des informations concernant l'échec d'une association de référentiels

L'`describe-repository-association`exemple suivant renvoie des informations sur une association de référentiel qui utilise un référentiel GitHub d'entreprise et qui est dans l'`Failed`état.

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RepositoryAssociation": {
        "ProviderType": "GitHubEnterpriseServer",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596217036.892,
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "State": "Failed",
        "StateReason": "Failed, Please retry.",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "Owner": "sample-owner"
    }
}
```

Pour plus d'informations, consultez la section Créer une association de référentiels de serveurs d' GitHub entreprise dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

Exemple 3 : pour renvoyer des informations sur une association de référentiels dissociante

L'`describe-repository-association`exemple suivant renvoie des informations sur une association de référentiel qui utilise un référentiel GitHub d'entreprise et qui est dans l'`Disassociating`état.

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RepositoryAssociation": {
        "ProviderType": "GitHubEnterpriseServer",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596217036.892,
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "State": "Disassociating",
        "StateReason": "Source code access removal in progress",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "Owner": "sample-owner"
    }
}
```

Pour plus d'informations, consultez la section Créer une association de référentiels de serveurs d' GitHub entreprise dans Amazon CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous DescribeRepositoryAssociationà la section Référence des AWS CLI commandes.

`disassociate-repository`

L'exemple de code suivant montre comment utiliser`disassociate-repository`.

AWS CLI

Pour dissocier une association de référentiels

Ce qui suit `disassociate-repository` dissocie une association de référentiel qui utilise un AWS CodeCommit référentiel.

```
aws codeguru-reviewer disassociate-repository \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RepositoryAssociation": {
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "my-repository",
        "Owner": "123456789012",
        "ProviderType": "CodeCommit",
        "State": "Disassociating",
        "LastUpdatedTimeStamp": 1618939174.759,
        "CreatedTimeStamp": 1595636947.096
    },
    "Tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

Pour plus d'informations, consultez Dissocier un référentiel dans CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous DisassociateRepositoryà la section Référence des AWS CLI commandes.

`list-code-reviews`

L'exemple de code suivant montre comment utiliser`list-code-reviews`.

AWS CLI

Pour répertorier les révisions de code créées sur votre AWS compte au cours des 90 derniers jours.

L'`list-code-reviews`exemple suivant répertorie les révisions de code créées au cours des 90 derniers jours à l'aide de pull requests.

```
aws codeguru-reviewer list-code-reviews \
    --type PullRequest
```

Sortie :

```
{
    "CodeReviewSummaries": [
        {
            "LastUpdatedTimeStamp": 1588897288.054,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProviderType": "GitHub",
            "PullRequestId": "5",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 24,
                "FindingsCount": 1
            },
            "CreatedTimeStamp": 1588897068.512,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588869793.263,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProviderType": "GitHub",
            "PullRequestId": "4",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 29,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588869575.949,
            "State": "Completed",
```

```
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588870511.211,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ProviderType": "GitHub",
            "PullRequestId": "4",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 2,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588870292.425,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588118522.452,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "ProviderType": "GitHub",
            "PullRequestId": "3",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 29,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588118301.131,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588112205.207,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "ProviderType": "GitHub",
```

```
            "PullRequestId": "2",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 25,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588111987.443,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588104489.981,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "ProviderType": "GitHub",
            "PullRequestId": "1",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 25,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588104270.223,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher toutes les révisions de code dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous ListCodeReviewsà la section Référence des AWS CLI commandes.

`list-recommendation-feedback`

L'exemple de code suivant montre comment utiliser`list-recommendation-feedback`.

AWS CLI

Pour répertorier les commentaires des clients concernant une recommandation sur un référentiel associé

Ce qui suit `list-recommendation-feedback` répertorie les commentaires des clients sur toutes les recommandations relatives à la révision d'un code. Cette révision du code contient un commentaire, un ThumbsUp « », de la part d'un client.

```
aws codeguru-reviewer list-recommendation-feedback \
    --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678
```

Sortie :

```
{
    "RecommendationFeedbackSummaries": [
        {
            "RecommendationId":
 "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
            "Reactions": [
                "ThumbsUp"
            ],
            "UserId": "aws-user-id"
        }
    ]
}
```

Pour plus d'informations, consultez l'étape 4 : fournir des commentaires dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous ListRecommendationFeedbackà la section Référence des AWS CLI commandes.

`list-recommendations`

L'exemple de code suivant montre comment utiliser`list-recommendations`.

AWS CLI

Pour répertorier les recommandations relatives à une révision de code terminée

L'`list-recommendations`exemple suivant répertorie les recommandations pour une révision de code complète. Cette révision du code comporte une recommandation.

```
aws codeguru-reviewer list-recommendations \
    --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RecommendationSummaries": [
        {
            "Description": "\n\n**Problem**  \n You are using a `ConcurrentHashMap`, but your usage of `containsKey()` and `get()` may not be thread-safe at lines: **63 and 64**. In between the check and the `get()` another thread can remove the key and the `get()` will return `null`. The remove that can remove the key is at line: **59**.\n\n**Fix**  \n Consider calling `get()`, checking instead of your current check if the returned object is `null`, and then using that object only, without calling `get()` again.\n\n**More info**  \n [View an example on GitHub](https://github.com/apache/hadoop/blob/f16cf877e565084c66bc63605659b157c4394dc8/hadoop-tools/hadoop-aws/src/main/java/org/apache/hadoop/fs/s3a/s3guard/S3Guard.java#L302-L304) (external link).",
            "RecommendationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "StartLine": 63,
            "EndLine": 64,
            "FilePath": "src/main/java/com/company/sample/application/CreateOrderThread.java"
        }
    ]
}
```

Pour plus d'informations, consultez l'étape 4 : fournir des commentaires dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous ListRecommendationsà la section Référence des AWS CLI commandes.

`list-repository-associations`

L'exemple de code suivant montre comment utiliser`list-repository-associations`.

AWS CLI

Pour répertorier les associations de référentiels dans votre AWS compte

L'`list-repository-associations`exemple suivant renvoie une liste des objets récapitulatifs des associations de référentiels présents dans votre compte. Vous pouvez filtrer la liste renvoyée par `ProviderTypeName`,`State`, et`Owner`.

```
aws codeguru-reviewer list-repository-associations
```

Sortie :

```
{
    "RepositoryAssociationSummaries": [
        {
            "LastUpdatedTimeStamp": 1595886609.616,
            "Name": "test",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Owner": "sample-owner",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProviderType": "Bitbucket"
        },
        {
            "LastUpdatedTimeStamp": 1595636969.035,
            "Name": "CodeDeploy-CodePipeline-ECS-Tutorial",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Owner": "123456789012",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProviderType": "CodeCommit"
        },
        {
            "LastUpdatedTimeStamp": 1595634785.983,
            "Name": "My-ecs-beta-repo",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Owner": "123456789012",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ProviderType": "CodeCommit"
```

```
        },
        {
            "LastUpdatedTimeStamp": 1590712811.77,
            "Name": "MyTestCodeCommit",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "Owner": "123456789012",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "ProviderType": "CodeCommit"
        },
        {
            "LastUpdatedTimeStamp": 1588102637.649,
            "Name": "aws-codeguru-profiler-sample-application",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "Owner": "sample-owner",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "ProviderType": "GitHub"
        },
        {
            "LastUpdatedTimeStamp": 1588028233.995,
            "Name": "codeguru-profiler-demo-app",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "Owner": "sample-owner",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "ProviderType": "GitHub"
        }
    ]
}
```

Pour plus d'informations, consultez Afficher toutes les associations de référentiels dans CodeGuru Reviewer dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous ListRepositoryAssociationsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un référentiel associé

La `list-tags-for-resource` liste suivante répertorie les balises d'un référentiel associé. Ce référentiel associé possède deux balises.

```
aws codeguru-reviewer list-tags-for-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "Tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

Pour plus d'informations, consultez la section Afficher les balises pour un référentiel associé à un CodeGuru réviseur (AWS CLI) dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-recommendation-feedback`

L'exemple de code suivant montre comment utiliser`put-recommendation-feedback`.

AWS CLI

Pour ajouter une recommandation à une révision de code

Ce qui suit `put-recommendation-feedback` contient une ThumbsUp recommandation concernant une révision du code.

```
aws codeguru-reviewer put-recommendation-feedback \
    --code-review-arn \arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
```

```
    --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \
    --reactions ThumbsUp
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez l'étape 4 : fournir des commentaires dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous PutRecommendationFeedbackà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une balise à un référentiel associé

Ce qui suit `tag-resource` ajoute deux balises à un référentiel associé

```
aws codeguru-reviewer tag-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tags Status=Secret,Team=Saanvi
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Ajouter une balise à un référentiel associé à un CodeGuru réviseur (AWS CLI) et Ajouter ou mettre à jour des balises pour un référentiel associé à un CodeGuru réviseur (AWS CLI) dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer le balisage d'un référentiel associé

Ce qui suit `untag-resource` supprime deux balises avec les clés « Secret » et « Team » d'un référentiel associé.

```
aws codeguru-reviewer untag-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tag-keys Status Team
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer les balises d'un référentiel associé à un CodeGuru réviseur (AWS CLI) dans le guide de l'utilisateur Amazon CodeGuru Reviewer.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

CodePipeline exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CodePipeline.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`acknowledge-job`

L'exemple de code suivant montre comment utiliser`acknowledge-job`.

AWS CLI

Pour récupérer des informations sur une tâche spécifiée

Cet exemple renvoie des informations sur une tâche spécifiée, y compris le statut de cette tâche s'il existe. Ceci n'est utilisé que pour les travailleurs et les actions personnalisées. Pour déterminer la valeur de nonce et l'ID de tâche, utilisez aws poll-for-jobs codepipeline.

Commande :

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

Sortie :

```
{
  "status": "InProgress"
}
```

- Pour plus de détails sur l'API, reportez-vous AcknowledgeJobà la section Référence des AWS CLI commandes.

`create-custom-action-type`

L'exemple de code suivant montre comment utiliser`create-custom-action-type`.

AWS CLI

Pour créer une action personnalisée

Cet exemple crée une action personnalisée pour AWS CodePipeline utiliser un fichier JSON déjà créé (ici nommé MyCustomAction .json) qui contient la structure de l'action personnalisée. Pour plus d'informations sur les conditions requises pour créer une action personnalisée, notamment sur la structure du fichier, consultez le guide de AWS CodePipeline l'utilisateur.

```
aws codepipeline create-custom-action-type --cli-input-json file://
MyCustomAction.json
```

Contenu du fichier JSON `MyCustomAction.json` :

```
{
    "category": "Build",
    "provider": "MyJenkinsProviderName",
    "version": "1",
    "settings": {
```

```
        "entityUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/",
        "executionUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/
lastSuccessfulBuild/{ExternalExecutionId}/"
    },
    "configurationProperties": [
        {
            "name": "MyJenkinsExampleBuildProject",
            "required": true,
            "key": true,
            "secret": false,
            "queryable": false,
            "description": "The name of the build project must be provided when this
 action is added to the pipeline.",
            "type": "String"
        }
    ],
    "inputArtifactDetails": {
        "maximumCount": 1,
        "minimumCount": 0
    },
    "outputArtifactDetails": {
        "maximumCount": 1,
        "minimumCount": 0
    }
}
```

Cette commande renvoie la structure de l'action personnalisée.

- Pour plus de détails sur l'API, reportez-vous CreateCustomActionTypeà la section Référence des AWS CLI commandes.

`create-pipeline`

L'exemple de code suivant montre comment utiliser`create-pipeline`.

AWS CLI

Pour créer un pipeline

Cet exemple crée un pipeline en AWS CodePipeline utilisant un fichier JSON déjà créé (ici nommé MySecondPipeline .json) qui contient la structure du pipeline. Pour plus d'informations sur les exigences relatives à la création d'un pipeline, notamment sur la structure du fichier, consultez le guide de AWS CodePipeline l'utilisateur.

Commande :

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

Exemple de contenu d'un fichier JSON :

```
{
 "pipeline": {
  "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
  "stages": [
    {
      "name": "Source",
      "actions": [
        {
          "inputArtifacts": [],
          "name": "Source",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "configuration": {
            "S3Bucket": "awscodepipeline-demo-bucket",
            "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
          },
          "runOrder": 1
        }
      ]
    },
    {
      "name": "Beta",
      "actions": [
        {
          "inputArtifacts": [
            {
              "name": "MyApp"
            }
```

```
          ],
          "name": "CodePipelineDemoFleet",
          "actionTypeId": {
            "category": "Deploy",
            "owner": "AWS",
            "version": "1",
            "provider": "CodeDeploy"
          },
          "outputArtifacts": [],
          "configuration": {
            "ApplicationName": "CodePipelineDemoApplication",
            "DeploymentGroupName": "CodePipelineDemoFleet"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "codepipeline-us-east-1-11EXAMPLE11"
  },
  "name": "MySecondPipeline",
  "version": 1
 }
}
```

Sortie :

```
This command returns the structure of the pipeline.
```

- Pour plus de détails sur l'API, reportez-vous CreatePipelineà la section Référence des AWS CLI commandes.

`delete-custom-action-type`

L'exemple de code suivant montre comment utiliser`delete-custom-action-type`.

AWS CLI

Pour supprimer une action personnalisée

Cet exemple supprime une action personnalisée en AWS CodePipeline utilisant un fichier JSON déjà créé (ici nommé DeleteMyCustomAction .json) qui contient le type d'action, le nom du fournisseur et le numéro de version de l'action à supprimer. Utilisez la list-action-types commande pour afficher les valeurs correctes pour la catégorie, la version et le fournisseur.

Commande :

```
aws codepipeline delete-custom-action-type --cli-input-json file://
DeleteMyCustomAction.json
```

Exemple de contenu d'un fichier JSON :

```
{
  "category": "Build",
  "version": "1",
  "provider": "MyJenkinsProviderName"
}
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous DeleteCustomActionTypeà la section Référence des AWS CLI commandes.

`delete-pipeline`

L'exemple de code suivant montre comment utiliser`delete-pipeline`.

AWS CLI

Pour supprimer un pipeline

Cet exemple supprime un pipeline nommé MySecondPipeline from AWS CodePipeline. Utilisez la commande list-pipelines pour afficher la liste des pipelines associés à votre AWS compte.

Commande :

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous DeletePipelineà la section Référence des AWS CLI commandes.

`delete-webhook`

L'exemple de code suivant montre comment utiliser`delete-webhook`.

AWS CLI

Pour supprimer un webhook

L'`delete-webhook`exemple suivant supprime un webhook pour une action source de GitHub version 1. Vous devez utiliser la `deregister-webhook-with-third-party` commande pour désenregistrer le webhook avant de le supprimer.

```
aws codepipeline delete-webhook \
    --name my-webhook
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer le webhook de votre GitHub source dans le guide de l'AWS CodePipeline utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteWebhookà la section Référence des AWS CLI commandes.

`deregister-webhook-with-third-party`

L'exemple de code suivant montre comment utiliser`deregister-webhook-with-third-party`.

AWS CLI

Pour désenregistrer un webhook

L'`deregister-webhook-with-third-party`exemple suivant supprime un webhook pour une action source de GitHub version 1. Vous devez annuler l'enregistrement du webhook avant de supprimer ce dernier.

```
aws codepipeline deregister-webhook-with-third-party \
    --webhook-name my-webhook
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer le webhook de votre GitHub source dans le guide de l'AWS CodePipeline utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeregisterWebhookWithThirdPartyà la section Référence des AWS CLI commandes.

`disable-stage-transition`

L'exemple de code suivant montre comment utiliser`disable-stage-transition`.

AWS CLI

Pour désactiver une transition vers une étape d'un pipeline

Cet exemple désactive les transitions vers la phase bêta du MyFirstPipeline pipeline dans AWS CodePipeline.

Commande :

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-
name Beta  --transition-type Inbound
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous DisableStageTransitionà la section Référence des AWS CLI commandes.

`enable-stage-transition`

L'exemple de code suivant montre comment utiliser`enable-stage-transition`.

AWS CLI

Pour permettre la transition vers une étape d'un pipeline

Cet exemple permet de passer à la phase bêta du MyFirstPipeline pipeline dans AWS CodePipeline.

Commande :

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, reportez-vous EnableStageTransitionà la section Référence des AWS CLI commandes.

`get-job-details`

L'exemple de code suivant montre comment utiliser`get-job-details`.

AWS CLI

Pour obtenir les détails d'une offre d'emploi

Cet exemple renvoie les détails d'une tâche dont l'ID est représenté par F4F4FF82-2D11-Example. Cette commande est uniquement utilisée pour les actions personnalisées. Lorsque cette commande est appelée, elle AWS CodePipeline renvoie des informations d'identification temporaires pour le compartiment Amazon S3 utilisé pour stocker les artefacts du pipeline, si nécessaire pour l'action personnalisée. Cette commande renverra également toutes les valeurs secrètes définies pour l'action, le cas échéant.

Commande :

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

Sortie :

```
{
 "jobDetails": {
  "accountId": "111111111111",
  "data": {
    "actionConfiguration": {
```

```
      "__type": "ActionConfiguration",
      "configuration": {
        "ProjectName": "MyJenkinsExampleTestProject"
      }
    },
    "actionTypeId": {
      "__type": "ActionTypeId",
      "category": "Test",
      "owner": "Custom",
      "provider": "MyJenkinsProviderName",
      "version": "1"
    },
    "artifactCredentials": {
      "__type": "AWSSessionCredentials",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
      "sessionToken":
 "fICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpEIbb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs
    },
    "inputArtifacts": [
      {
        "__type": "Artifact",
        "location": {
          "s3Location": {
            "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
            "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
          },
          "type": "S3"
        },
        "name": "MyAppBuild"
      }
    ],
    "outputArtifacts": [],
    "pipelineContext": {
      "__type": "PipelineContext",
      "action": {
        "name": "MyJenkinsTest-Action"
      },
      "pipelineName": "MySecondPipeline",
      "stage": {
```

```
        "name": "Testing"
      }
    }
  },
  "id": "f4f4ff82-2d11-EXAMPLE"
 }
}
```

- Pour plus de détails sur l'API, reportez-vous GetJobDetailsà la section Référence des AWS CLI commandes.

`get-pipeline-state`

L'exemple de code suivant montre comment utiliser`get-pipeline-state`.

AWS CLI

Pour obtenir des informations sur l'état d'un pipeline

Cet exemple renvoie l'état le plus récent d'un pipeline nommé MyFirstPipeline.

Commande :

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

Sortie :

```
{
 "created": 1446137312.204,
 "pipelineName": "MyFirstPipeline",
 "pipelineVersion": 1,
 "stageStates": [
  {
    "actionStates": [
      {
        "actionName": "Source",
        "entityUrl": "https://console.aws.amazon.com/s3/home?#",
        "latestExecution": {
          "lastStatusChange": 1446137358.328,
          "status": "Succeeded"
        }
```

```
      }
    ],
    "stageName": "Source"
  },
  {
    "actionStates": [
      {
        "actionName": "CodePipelineDemoFleet",
        "entityUrl": "https://console.aws.amazon.com/codedeploy/home?#/applications/
CodePipelineDemoApplication/deployment-groups/CodePipelineDemoFleet",
        "latestExecution": {
          "externalExecutionId": "d-EXAMPLE",
          "externalExecutionUrl": "https://console.aws.amazon.com/codedeploy/home?#/
deployments/d-EXAMPLE",
          "lastStatusChange": 1446137493.131,
          "status": "Succeeded",
          "summary": "Deployment Succeeded"
        }
      }
    ],
    "inboundTransitionState": {
      "enabled": true
    },
    "stageName": "Beta"
  }
 ],
 "updated": 1446137312.204
}
```

- Pour plus de détails sur l'API, reportez-vous GetPipelineStateà la section Référence des AWS CLI commandes.

`get-pipeline`

L'exemple de code suivant montre comment utiliser`get-pipeline`.

AWS CLI

Pour afficher la structure d'un pipeline

Cet exemple renvoie la structure d'un pipeline nommé MyFirstPipeline.

Commande :

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

Sortie :

```
{
  "pipeline": {
      "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
      "stages": [
          {
              "name": "Source",
              "actions": [
                  {
                      "inputArtifacts": [],
                      "name": "Source",
                      "actionTypeId": {
                          "category": "Source",
                          "owner": "AWS",
                          "version": "1",
                          "provider": "S3"
                      },
                      "outputArtifacts": [
                          {
                              "name": "MyApp"
                          }
                      ],
                      "configuration": {
                          "S3Bucket": "awscodepipeline-demo-bucket",
                          "S3ObjectKey": "aws-codepipeline-s3-aws-
codedeploy_linux.zip"
                      },
                      "runOrder": 1
                  }
              ]
          },
          {
              "name": "Beta",
              "actions": [
                  {
                      "inputArtifacts": [
                          {
                              "name": "MyApp"
                          }
                      ],
```

```
                    "name": "CodePipelineDemoFleet",
                    "actionTypeId": {
                        "category": "Deploy",
                        "owner": "AWS",
                        "version": "1",
                        "provider": "CodeDeploy"
                    },
                    "outputArtifacts": [],
                    "configuration": {
                        "ApplicationName": "CodePipelineDemoApplication",
                        "DeploymentGroupName": "CodePipelineDemoFleet"
                    },
                    "runOrder": 1
                }
            ]
        }
    ],
    "artifactStore": {
        "type": "S3",
        "location": "codepipeline-us-east-1-11EXAMPLE11"
    },
    "name": "MyFirstPipeline",
    "version": 1
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPipelineà la section Référence des AWS CLI commandes.

`list-action-executions`

L'exemple de code suivant montre comment utiliser`list-action-executions`.

AWS CLI

Pour répertorier les exécutions d'actions

L'`list-action-executions`exemple suivant affiche les détails de l'exécution d'une action pour un pipeline, tels que l'ID d'exécution de l'action, les artefacts d'entrée, les artefacts de sortie, le résultat de l'exécution et le statut.

```
aws codepipeline list-action-executions \
```

```
    --pipeline-name myPipeline
```

Sortie :

```
{
    "actionExecutionDetails": [
        {
            "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
            "actionExecutionId": "EXAMPLE4-2ee8-4853-bd6a-111111158148",
            "pipelineVersion": 12,
            "stageName": "Deploy",
            "actionName": "Deploy",
            "startTime": 1598572628.6,
            "lastUpdateTime": 1598572661.255,
            "status": "Succeeded",
            "input": {
                "actionTypeId": {
                    "category": "Deploy",
                    "owner": "AWS",
                    "provider": "CodeDeploy",
                    "version": "1"
                },
                "configuration": {
                    "ApplicationName": "my-application",
                    "DeploymentGroupName": "my-deployment-group"
                },
                "resolvedConfiguration": {
                    "ApplicationName": "my-application",
                    "DeploymentGroupName": "my-deployment-group"
                },
                "region": "us-east-1",
                "inputArtifacts": [
                    {
                        "name": "SourceArtifact",
                        "s3location": {
                            "bucket": "artifact-bucket",
                            "key": "myPipeline/SourceArti/key"
                        }
                    }
                ],
                "namespace": "DeployVariables"
            },
            "output": {
```

```
                "outputArtifacts": [],
                "executionResult": {
                    "externalExecutionId": "d-EXAMPLEE5",
                    "externalExecutionSummary": "Deployment Succeeded",
                    "externalExecutionUrl": "https://myaddress.com"
                },
                "outputVariables": {}
            }
        },
        {
            "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
            "actionExecutionId": "EXAMPLE5-abb4-4192-9031-11111113a7b0",
            "pipelineVersion": 12,
            "stageName": "Source",
            "actionName": "Source",
            "startTime": 1598572624.387,
            "lastUpdateTime": 1598572628.16,
            "status": "Succeeded",
            "input": {
                "actionTypeId": {
                    "category": "Source",
                    "owner": "AWS",
                    "provider": "CodeCommit",
                    "version": "1"
                },
                "configuration": {
                    "BranchName": "production",
                    "PollForSourceChanges": "false",
                    "RepositoryName": "my-repo"
                },
                "resolvedConfiguration": {
                    "BranchName": "production",
                    "PollForSourceChanges": "false",
                    "RepositoryName": "my-repo"
                },
                "region": "us-east-1",
                "inputArtifacts": [],
                "namespace": "SourceVariables"
            },
            "output": {
                "outputArtifacts": [
                    {
                        "name": "SourceArtifact",
                        "s3location": {
```

```
                            "bucket": "amzn-s3-demo-bucket",
                            "key": "myPipeline/SourceArti/key"
                        }
                    }
                ],
                "executionResult": {
                    "externalExecutionId":
 "1111111ad99dcd35914c00b7fbea13995EXAMPLE",
                    "externalExecutionSummary": "Edited template.yml",
                    "externalExecutionUrl": "https://myaddress.com"
                },
                "outputVariables": {
                    "AuthorDate": "2020-05-08T17:45:43Z",
                    "BranchName": "production",
                    "CommitId": "EXAMPLEad99dcd35914c00b7fbea139951111111",
                    "CommitMessage": "Edited template.yml",
                    "CommitterDate": "2020-05-08T17:45:43Z",
                    "RepositoryName": "my-repo"
                }
            }
        },
. . . .
```

Pour plus d'informations, consultez la section Afficher les exécutions d'actions (CLI) dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListActionExecutionsà la section Référence des AWS CLI commandes.

`list-action-types`

L'exemple de code suivant montre comment utiliser`list-action-types`.

AWS CLI

Pour afficher les types d'actions disponibles

Utilisée seule, la list-action-types commande renvoie la structure de toutes les actions disponibles pour votre AWS compte. Cet exemple utilise l'action-owner-filter option -- pour renvoyer uniquement des actions personnalisées.

Commande :

```
aws codepipeline list-action-types --action-owner-filter Custom
```

Sortie :

```
{
  "actionTypes": [
      {
          "inputArtifactDetails": {
              "maximumCount": 5,
              "minimumCount": 0
          },
          "actionConfigurationProperties": [
              {
                  "secret": false,
                  "required": true,
                  "name": "MyJenkinsExampleBuildProject",
                  "key": true,
                  "queryable": true
              }
          ],
          "outputArtifactDetails": {
              "maximumCount": 5,
              "minimumCount": 0
          },
          "id": {
              "category": "Build",
              "owner": "Custom",
              "version": "1",
              "provider": "MyJenkinsProviderName"
          },
          "settings": {
              "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
              "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
          }
      },
      {
          "inputArtifactDetails": {
              "maximumCount": 5,
              "minimumCount": 0
          },
          "actionConfigurationProperties": [
              {
```

```
                    "secret": false,
                    "required": true,
                    "name": "MyJenkinsExampleTestProject",
                    "key": true,
                    "queryable": true
                }
            ],
            "outputArtifactDetails": {
                "maximumCount": 5,
                "minimumCount": 0
            },
            "id": {
                "category": "Test",
                "owner": "Custom",
                "version": "1",
                "provider": "MyJenkinsProviderName"
            },
            "settings": {
                "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
                "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListActionTypesà la section Référence des AWS CLI commandes.

`list-pipeline-executions`

L'exemple de code suivant montre comment utiliser`list-pipeline-executions`.

AWS CLI

Pour consulter l'historique d'exécution du pipeline

L'`list-pipeline-executions`exemple suivant montre l'historique d'exécution d'un pipeline dans votre AWS compte.

```
aws codepipeline list-pipeline-executions \
    --pipeline-name MyPipeline
```

Sortie :

```
{
    "pipelineExecutionSummaries": [
        {
            "lastUpdateTime": 1496380678.648,
            "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
            "startTime": 1496380258.243,
            "status": "Succeeded"
        },
        {
            "lastUpdateTime": 1496591045.634,
            "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d84l-d7eu3EXAMPLE",
            "startTime": 1496590401.222,
            "status": "Succeeded"
        },
        {
            "lastUpdateTime": 1496946071.6456,
            "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
            "startTime": 1496945471.5645,
            "status": "Succeeded"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher l'historique des exécutions dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListPipelineExecutionsà la section Référence des AWS CLI commandes.

`list-pipelines`

L'exemple de code suivant montre comment utiliser`list-pipelines`.

AWS CLI

Pour consulter la liste des pipelines

Cet exemple répertorie tous les AWS CodePipeline pipelines associés au AWS compte de l'utilisateur.

Commande :

```
aws codepipeline list-pipelines
```

Sortie :

```
{
  "pipelines": [
      {
          "updated": 1439504274.641,
          "version": 1,
          "name": "MyFirstPipeline",
          "created": 1439504274.641
      },
      {
          "updated": 1436461837.992,
          "version": 2,
          "name": "MySecondPipeline",
          "created": 1436460801.381
      }
      ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPipelinesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags

L'`list-tags-for-resource`exemple suivant extrait une liste de toutes les balises associées à la ressource de pipeline spécifiée.

```
aws codepipeline list-tags-for-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

Sortie :

```
{
```

```
    "tags": {
        "Project": "ProjectA",
        "IscontainerBased": "true"
    }
}
```

Pour plus d'informations, voir Afficher les balises d'un pipeline (CLI) dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-webhooks`

L'exemple de code suivant montre comment utiliser`list-webhooks`.

AWS CLI

Pour répertorier les webhooks

L'`list-webhooks`exemple suivant extrait une liste de toutes les balises associées à la ressource de pipeline spécifiée.

```
aws codepipeline list-webhooks \
    --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \
    --region "eu-central-1"
```

Sortie :

```
{
    "webhooks": [
        {
            "url": "https://webhooks.domain.com/
trigger111111111EXAMPLE1111111111111111": {
                "authenticationConfiguration": {
                    "SecretToken": "Secret"
                },
                "name": "my-webhook",
                "authentication": "GITHUB_HMAC",
                "targetPipeline": "my-Pipeline",
                "targetAction": "Source",
```

```
                "filters": [
                    {
                        "jsonPath": "$.ref",
                        "matchEquals": "refs/heads/{Branch}"
                    }
                ]
            },
            "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-
webhook"
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les webhooks de votre compte dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListWebhooksà la section Référence des AWS CLI commandes.

`poll-for-jobs`

L'exemple de code suivant montre comment utiliser`poll-for-jobs`.

AWS CLI

Pour consulter les offres d'emploi disponibles

Cet exemple renvoie des informations sur toutes les tâches sur lesquelles un travailleur doit agir. Cet exemple utilise un fichier JSON prédéfini (MyActionTypeInfo.json) pour fournir des informations sur le type d'action pour lequel le travailleur traite les tâches. Cette commande est uniquement utilisée pour les actions personnalisées. Lorsque cette commande est appelée, elle AWS CodePipeline renvoie des informations d'identification temporaires pour le compartiment Amazon S3 utilisé pour stocker les artefacts du pipeline. Cette commande renverra également toutes les valeurs secrètes définies pour l'action, le cas échéant.

Commande :

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

Exemple de contenu d'un fichier JSON :

```
{
  "actionTypeId": {
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
      "ProjectName": "MyJenkinsTestProject"
  }
}
```

Sortie :

```
{
 "jobs": [
  {
    "accountId": "111111111111",
    "data": {
      "actionConfiguration": {
        "__type": "ActionConfiguration",
        "configuration": {
          "ProjectName": "MyJenkinsExampleTestProject"
        }
      },
      "actionTypeId": {
        "__type": "ActionTypeId",
        "category": "Test",
        "owner": "Custom",
        "provider": "MyJenkinsProviderName",
        "version": "1"
      },
      "artifactCredentials": {
        "__type": "AWSSessionCredentials",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "sessionToken":
 "fICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpEIbb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs
```

```
      },
      "inputArtifacts": [
        {
          "__type": "Artifact",
          "location": {
            "s3Location": {
              "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
              "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
            },
            "type": "S3"
          },
          "name": "MyAppBuild"
        }
      ],
      "outputArtifacts": [],
      "pipelineContext": {
        "__type": "PipelineContext",
        "action": {
          "name": "MyJenkinsTest-Action"
        },
        "pipelineName": "MySecondPipeline",
        "stage": {
          "name": "Testing"
        }
      }
    },
    "id": "ef66c259-64f9-EXAMPLE",
    "nonce": "3"
  }
 ]
}
```

- Pour plus de détails sur l'API, reportez-vous PollForJobsà la section Référence des AWS CLI commandes.

`put-webhook`

L'exemple de code suivant montre comment utiliser`put-webhook`.

AWS CLI

Pour créer un webhook

L'`put-webhook`exemple suivant crée un webhook pour une action source de GitHub version 1. Après avoir créé le webhook, vous devez utiliser la commande register-webhook-with-third -party pour l'enregistrer.

```
aws codepipeline put-webhook \
    --cli-input-json file://webhook_json.json \
    --region "eu-central-1"
```

Contenu de `webhook_json.json` :

```
{
    "webhook": {
        "name": "my-webhook",
        "targetPipeline": "pipeline_name",
        "targetAction": "source_action_name",
        "filters": [
            {
                "jsonPath": "$.ref",
                "matchEquals": "refs/heads/{Branch}"
            }
        ],
        "authentication": "GITHUB_HMAC",
        "authenticationConfiguration": {
            "SecretToken": "secret"
        }
    }
}
```

Sortie :

```
{
    "webhook": {
        "url": "https://webhooks.domain.com/
trigger111111111EXAMPLE11111111111111111",
        "definition": {
            "authenticationConfiguration": {
                "SecretToken": "secret"
            },
            "name": "my-webhook",
            "authentication": "GITHUB_HMAC",
            "targetPipeline": "pipeline_name",
            "targetAction": "Source",
```

```
            "filters": [
                {
                    "jsonPath": "$.ref",
                    "matchEquals": "refs/heads/{Branch}"
                }
            ]
        },
        "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-webhook"
    },
    "tags": [
        {
            "key": "Project",
            "value": "ProjectA"
        }
    ]
}
```

Pour plus d'informations, voir Création d'un webhook pour une GitHub source dans le guide de l'AWS CodePipeline utilisateur.

- Pour plus de détails sur l'API, reportez-vous PutWebhookà la section Référence des AWS CLI commandes.

`retry-stage-execution`

L'exemple de code suivant montre comment utiliser`retry-stage-execution`.

AWS CLI

Pour réessayer une action qui a échoué

L'`retry-stage-execution`exemple suivant réessaie une étape dont l'action a échoué.

```
aws codepipeline retry-stage-execution \
    --pipeline-name MyPipeline \
    --stage-name Deploy \
    --pipeline-execution-id b59babff-5f34-EXAMPLE \
    --retry-mode FAILED_ACTIONS
```

Sortie :

```
{
```

```
    "pipelineExecutionId": "b59babff-5f34-EXAMPLE"
}
```

Pour plus d'informations, consultez la section Réessayer les actions ayant échoué (CLI) dans le guide de l'AWS CodePipeline utilisateur.

- Pour plus de détails sur l'API, reportez-vous RetryStageExecutionà la section Référence des AWS CLI commandes.

`start-pipeline-execution`

L'exemple de code suivant montre comment utiliser`start-pipeline-execution`.

AWS CLI

Pour exécuter la dernière révision via un pipeline

Cet exemple exécute la dernière révision présente dans l'étape source d'un pipeline via le pipeline nommé « MyFirstPipeline ».

Commande :

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

Sortie :

```
{
  "pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous StartPipelineExecutionà la section Référence des AWS CLI commandes.

`stop-pipeline-execution`

L'exemple de code suivant montre comment utiliser`stop-pipeline-execution`.

AWS CLI

Pour arrêter l'exécution d'un pipeline

L'`stop-pipeline-execution`exemple suivant montre par défaut comment attendre la fin des actions en cours, puis arrête le pipeline. Vous ne pouvez pas choisir d'arrêter et d'attendre si l'exécution est déjà à l'état Stopping. Vous pouvez choisir d'arrêter et d'abandonner une exécution déjà à l'état Stopping.

```
aws codepipeline stop-pipeline-execution \
    --pipeline-name MyFirstPipeline \
    --pipeline-execution-id d-EXAMPLE \
    --reason "Stopping pipeline after the build action is done"
```

Cette commande ne renvoie aucun résultat.

Pour plus d'informations, voir Arrêter l'exécution d'un pipeline (CLI) dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StopPipelineExecutionà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant associe un ensemble de balises fournies à un pipeline. Utilisez cette commande pour ajouter ou modifier des balises.

```
aws codepipeline tag-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \
    --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des balises à un pipeline (CLI) dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des AWS balises d'une ressource de connexion

L'`untag-resource`exemple suivant supprime une balise de la ressource spécifiée.

```
aws codepipeline untag-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \
    --tag-keys Project IscontainerBased
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer les balises d'un pipeline (CLI) dans le guide de AWS CodePipeline l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-pipeline`

L'exemple de code suivant montre comment utiliser`update-pipeline`.

AWS CLI

Pour mettre à jour la structure d'un pipeline

Cet exemple utilise la commande update-pipeline avec l'argument --cli-input-json . Cet exemple utilise un fichier JSON prédéfini (MyFirstPipeline.json) pour mettre à jour la structure d'un pipeline. AWS CodePipeline reconnaît le nom du pipeline contenu dans le fichier JSON, puis applique les modifications apportées aux champs modifiés de la structure du pipeline pour mettre à jour le pipeline.

Respectez les consignes suivantes lors de la création du fichier JSON prédéfini :

Si vous travaillez avec une structure de pipeline récupérée à l'aide de la commande get-pipeline, vous devez supprimer la section de métadonnées de la structure de pipeline dans le fichier JSON (les lignes « metadata » : {} et les champs « created », « pipelineRain » et « updated » qu'ils contiennent). Le nom du pipeline ne peut pas être modifié.

Commande :

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

Exemple de contenu de fichier JSON :

```
{
 "pipeline": {
  "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
  "stages": [
    {
      "name": "Source",
      "actions": [
        {
          "inputArtifacts": [],
          "name": "Source",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "configuration": {
            "S3Bucket": "awscodepipeline-demo-bucket2",
            "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
          },
          "runOrder": 1
        }
      ]
    },
    {
      "name": "Beta",
      "actions": [
        {
          "inputArtifacts": [
            {
              "name": "MyApp"
            }
```

```
          ],
          "name": "CodePipelineDemoFleet",
          "actionTypeId": {
            "category": "Deploy",
            "owner": "AWS",
            "version": "1",
            "provider": "CodeDeploy"
          },
          "outputArtifacts": [],
          "configuration": {
            "ApplicationName": "CodePipelineDemoApplication",
            "DeploymentGroupName": "CodePipelineDemoFleet"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "codepipeline-us-east-1-11EXAMPLE11"
  },
  "name": "MyFirstPipeline",
  "version": 1
 }
}
```

Sortie :

```
{
 "pipeline": {
  "artifactStore": {
    "location": "codepipeline-us-east-1-11EXAMPLE11",
    "type": "S3"
  },
  "name": "MyFirstPipeline",
  "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
  "stages": [
    {
      "actions": [
        {
          "actionTypeId": {
            "__type": "ActionTypeId",
```

```
            "category": "Source",
            "owner": "AWS",
            "provider": "S3",
            "version": "1"
          },
          "configuration": {
            "S3Bucket": "awscodepipeline-demo-bucket2",
            "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
          },
          "inputArtifacts": [],
          "name": "Source",
          "outputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "runOrder": 1
        }
      ],
      "name": "Source"
    },
    {
      "actions": [
        {
          "actionTypeId": {
            "__type": "ActionTypeId",
            "category": "Deploy",
            "owner": "AWS",
            "provider": "CodeDeploy",
            "version": "1"
          },
          "configuration": {
            "ApplicationName": "CodePipelineDemoApplication",
            "DeploymentGroupName": "CodePipelineDemoFleet"
          },
          "inputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "name": "CodePipelineDemoFleet",
          "outputArtifacts": [],
          "runOrder": 1
        }
```

```
      ],
      "name": "Beta"
    }
  ],
  "version": 3
 }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdatePipelineà la section Référence des AWS CLI commandes.

AWS CodeStar Exemples de notifications utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide AWS CodeStar des notifications.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-notification-rule`

L'exemple de code suivant montre comment utiliser`create-notification-rule`.

AWS CLI

Pour créer une règle de notification

L'`create-notification-rule`exemple suivant utilise un fichier JSON nommé `rule.json` pour créer une règle de notification nommée `MyNotificationRule` pour un référentiel nommé `MyDemoRepo` dans le AWS compte spécifié. Les notifications avec le type de `FULL` détail sont

envoyées à la rubrique Amazon SNS cible spécifiée lorsque des branches et des balises sont créées.

```
aws codestar-notifications create-notification-rule \
    --cli-input-json file://rule.json
```

Contenu de `rule.json` :

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

Sortie :

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Pour plus d'informations, consultez la section Création d'une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous CreateNotificationRuleà la section Référence des AWS CLI commandes.

`delete-notification-rule`

L'exemple de code suivant montre comment utiliser`delete-notification-rule`.

AWS CLI

Pour supprimer une règle de notification

L'`delete-notification-rule`exemple suivant supprime la règle de notification spécifiée.

```
aws codestar-notifications delete-notification-rule \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE
```

Sortie :

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Pour plus d'informations, voir Supprimer une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous DeleteNotificationRuleà la section Référence des AWS CLI commandes.

`delete-target`

L'exemple de code suivant montre comment utiliser`delete-target`.

AWS CLI

Pour supprimer une cible de règle de notification

L'`delete-target`exemple suivant supprime la cible spécifiée de toutes les règles de notification configurées pour l'utiliser comme cible, puis supprime la cible.

```
aws codestar-notifications  delete-target \
    --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \
    --force-unsubscribe-all
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer une cible de règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous DeleteTargetà la section Référence des AWS CLI commandes.

`describe-notification-rule`

L'exemple de code suivant montre comment utiliser`describe-notification-rule`.

AWS CLI

Pour récupérer les détails d'une règle de notification

L'`describe-notification-rule`exemple suivant récupère les détails de la règle de notification spécifiée.

```
aws codestar-notifications describe-notification-rule \
    --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/
dc82df7a-EXAMPLE
```

Sortie :

```
{
    "LastModifiedTimestamp": 1569199844.857,
    "EventTypes": [
        {
            "ServiceName": "CodeCommit",
            "EventTypeName": "Branches and tags: Created",
            "ResourceType": "Repository",
            "EventTypeId": "codecommit-repository-branches-and-tags-created"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL",
    "Resource": "arn:aws:codecommit:us-west-2:123456789012:MyDemoRepo",
    "Arn": "arn:aws:codestar-notifications:us-west-w:123456789012:notificationrule/
dc82df7a-EXAMPLE",
    "Targets": [
        {
            "TargetStatus": "ACTIVE",
            "TargetAddress": "arn:aws:sns:us-
west-2:123456789012:MyNotificationTopic",
            "TargetType": "SNS"
        }
```

```
    ],
    "Name": "MyNotificationRule",
    "CreatedTimestamp": 1569199844.857,
    "CreatedBy": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

Pour plus d'informations, consultez la section Afficher les règles de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous DescribeNotificationRuleà la section Référence des AWS CLI commandes.

`list-event-types`

L'exemple de code suivant montre comment utiliser`list-event-types`.

AWS CLI

Pour obtenir la liste des types d'événements pour une règle de notification

L'`list-event-types`exemple suivant extrait une liste filtrée de tous les types d'événements de notification disponibles pour les CodeDeploy applications. Si vous n'utilisez aucun filtre, la commande renvoie tous les types d'événements de notification pour tous les types de ressources.

```
aws codestar-notifications list-event-types \
    --filters Name=SERVICE_NAME,Value=CodeDeploy
```

Sortie :

```
{
    "EventTypes": [
        {
            "EventTypeId": "codedeploy-application-deployment-succeeded",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Succeeded",
            "ResourceType": "Application"
        },
        {
            "EventTypeId": "codedeploy-application-deployment-failed",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Failed",
            "ResourceType": "Application"
```

```
        },
        {
            "EventTypeId": "codedeploy-application-deployment-started",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Started",
            "ResourceType": "Application"
        }
    ]
}
```

Pour plus d'informations, voir Création d'une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListEventTypesà la section Référence des AWS CLI commandes.

`list-notification-rules`

L'exemple de code suivant montre comment utiliser`list-notification-rules`.

AWS CLI

Pour récupérer la liste des règles de notification

L'`list-notification-rules`exemple suivant extrait une liste de toutes les règles de notification dans la AWS région spécifiée.

```
aws codestar-notifications list-notification-rules --region us-east-1
```

Sortie :

```
{
    "NotificationRules": [
        {
            "Id": "dc82df7a-EXAMPLE",
            "Arn": "arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
        },
        {
            "Id": "8d1f0983-EXAMPLE",
            "Arn": "arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/8d1f0983-EXAMPLE"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les règles de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListNotificationRulesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour obtenir la liste des balises associées à une règle de notification

L'`list-tags-for-resource`exemple suivant extrait une liste de toutes les balises associées à la règle de notification spécifiée. Dans cet exemple, aucune balise n'est actuellement associée à la règle de notification.

```
aws codestar-notifications list-tags-for-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE
```

Sortie :

```
{
    "Tags": {}
}
```

Pour plus d'informations, voir Création d'une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-targets`

L'exemple de code suivant montre comment utiliser`list-targets`.

AWS CLI

Pour récupérer une liste des cibles des règles de notification

L'`list-targets`exemple suivant extrait une liste de toutes les cibles des règles de notification dans la AWS région spécifiée.

```
aws codestar-notifications list-targets \
    --region us-east-1
```

Sortie :

```
{
    "Targets": [
        {
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MySNSTopicForNotificationRules",
            "TargetType": "SNS",
            "TargetStatus": "ACTIVE"
        },
        {
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo",
            "TargetType": "SNS",
            "TargetStatus": "ACTIVE"
        }
    ]
}
```

Pour plus d'informations, voir Afficher les cibles des règles de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListTargetsà la section Référence des AWS CLI commandes.

`subscribe`

L'exemple de code suivant montre comment utiliser`subscribe`.

AWS CLI

Pour ajouter une cible à une règle de notification

L'`subscribe`exemple suivant ajoute une rubrique Amazon SNS en tant que cible pour la règle de notification spécifiée.

```
aws codestar-notifications subscribe \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE \
    --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

Sortie :

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Pour plus d'informations, consultez la section Ajouter ou supprimer une rubrique Amazon SNS en tant que cible pour une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus d'informations sur l'API, consultez Subscribe dans la Référence des commandes AWS CLI .

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter un tag à une règle de notification

L'`tag-resource`exemple suivant ajoute une balise avec le nom de clé `Team` et la valeur de `Li_Juan` à la règle de notification spécifiée.

```
aws codestar-notifications tag-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE \
    --tags Team=Li_Juan
```

Sortie :

```
{
    "Tags": {
        "Team": "Li_Juan"
    }
}
```

Pour plus d'informations, voir Création d'une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`unsubscribe`

L'exemple de code suivant montre comment utiliser`unsubscribe`.

AWS CLI

Pour supprimer une cible d'une règle de notification

L'`unsubscribe`exemple suivant supprime une rubrique Amazon SNS en tant que cible de la règle de notification spécifiée.

```
aws codestar-notifications unsubscribe \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE \
    --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

Sortie :

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
    "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
}
```

Pour plus d'informations, consultez la section Ajouter ou supprimer une rubrique Amazon SNS en tant que cible pour une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus d'informations sur l'API, consultez Unsubscribe dans la Référence des commandes AWS CLI .

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une règle de notification

L'`untag-resource`exemple suivant supprime la balise portant le nom `Team` de la clé de la règle de notification spécifiée.

```
aws codestar-notifications untag-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE \
    --tag-keys Team
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Modifier une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-notification-rule`

L'exemple de code suivant montre comment utiliser`update-notification-rule`.

AWS CLI

Pour mettre à jour une règle de notification

L'`update-notification-rule`exemple suivant met à jour une règle de notification nommée `MyNotificationRule` dans le AWS compte à `123456789012` l'aide d'un fichier JSON nommé`update.json`.

```
aws codestar-notifications update-notification-rule \
    --cli-input-json file://update.json
```

Contenu de `update.json` :

```
{
    "Name": "MyUpdatedNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

Sortie :

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Pour plus d'informations, voir Modifier une règle de notification dans le guide de l'utilisateur de la console AWS Developer Tools.

- Pour plus de détails sur l'API, reportez-vous UpdateNotificationRuleà la section Référence des AWS CLI commandes.

CodeConnections exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with CodeConnections.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-connection`

L'exemple de code suivant montre comment utiliser`create-connection`.

AWS CLI

Pour créer une connexion

L'`create-connection`exemple suivant montre comment créer une connexion à un référentiel tiers. Cet exemple crée une connexion dont le fournisseur tiers est Bitbucket.

Une connexion créée via la AWS CLI ou dont le statut AWS CloudFormation est En attente par défaut. Après avoir créé une connexion avec la CLI AWS CloudFormation, utilisez la console pour modifier la connexion afin de rendre son statut disponible.

```
aws codestar-connections create-connection \
    --provider-type Bitbucket \
    --connection-name MyConnection
```

Sortie :

```
{
    "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
}
```

Pour plus d'informations, consultez la section Création d'une connexion dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous CreateConnectionà la section Référence des AWS CLI commandes.

`create-host`

L'exemple de code suivant montre comment utiliser`create-host`.

AWS CLI

Pour créer un hôte

L'`create-host`exemple suivant montre comment créer un hôte pour représenter le point de terminaison de l'infrastructure sur laquelle votre fournisseur tiers est installé. Cet exemple crée un hôte sur lequel le fournisseur tiers installé est GitHub Enterprise Server.

Un hôte créé via la AWS CLI est en attente par défaut. Après avoir créé un hôte à l'aide de la CLI, utilisez la console ou l'interface de ligne de commande pour configurer l'hôte afin qu'il rende son statut disponible.

```
aws codestar-connections create-host \
    --name MyHost \
    --provider-type GitHubEnterpriseServer \
    --provider-endpoint "https://my-instance.dev"
```

Sortie :

```
{
    "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605"
}
```

Pour plus d'informations, consultez la section Créer un hôte (CLI) dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous CreateHostà la section Référence des AWS CLI commandes.

`delete-connection`

L'exemple de code suivant montre comment utiliser`delete-connection`.

AWS CLI

Pour supprimer une connexion

L'`delete-connection`exemple suivant montre comment supprimer une connexion.

```
aws codestar-connections delete-connection \
    --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer une connexion (CLI) dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous DeleteConnectionà la section Référence des AWS CLI commandes.

`delete-host`

L'exemple de code suivant montre comment utiliser`delete-host`.

AWS CLI

Pour supprimer un hôte

L'`delete-host`exemple suivant montre comment supprimer un hôte. Avant de pouvoir supprimer un hôte, vous devez supprimer toutes les connexions associées à l'hôte.

```
aws codestar-connections delete-host \
    --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-Host-28aef605"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer un hôte (CLI) dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous DeleteHostà la section Référence des AWS CLI commandes.

`get-connection`

L'exemple de code suivant montre comment utiliser`get-connection`.

AWS CLI

Pour obtenir des informations sur une connexion

L'`get-connection`exemple suivant montre les détails d'une connexion.

```
aws codestar-connections get-connection \
    --connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Sortie :

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "ProviderType": "Bitbucket",
        "OwnerAccountId": "123456789012",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

Pour plus d'informations, voir Afficher les détails de connexion dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous GetConnectionà la section Référence des AWS CLI commandes.

`get-host`

L'exemple de code suivant montre comment utiliser`get-host`.

AWS CLI

Pour obtenir des informations sur un hôte

L'`get-host`exemple suivant montre les détails d'un hôte :

```
aws codestar-connections get-host \
    --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/MyHost-28aef605
```

Sortie :

```
{
    "Name": "MyHost",
    "Status": "AVAILABLE",
    "ProviderType": "GitHubEnterpriseServer",
    "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

Pour plus d'informations, consultez la section Afficher les détails de l'hôte (CLI) dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous GetHostà la section Référence des AWS CLI commandes.

`list-connections`

L'exemple de code suivant montre comment utiliser`list-connections`.

AWS CLI

Pour répertorier les connexions

L'`list-connections`exemple suivant extrait une liste de toutes les connexions de votre compte pour le type de fournisseur Bitbucket. :

```
aws codestar-connections list-connections \
--provider-type Bitbucket \
--max-results 5 \
--next-token: next-token
```

Sortie :

```
{
    "Connections": [
        {
            "ConnectionName": "my-connection",
            "ProviderType": "Bitbucket",
            "Status": "PENDING",
            "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
            "OwnerAccountId": "123456789012"
```

```
        },
        {
            "ConnectionName": "my-other-connection",
            "ProviderType": "Bitbucket",
            "Status": "AVAILABLE",
            "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
            "OwnerAccountId": "123456789012"
        },
    ],
    "NextToken": "next-token"
}
```

Pour plus d'informations, consultez la section Lister les connexions (CLI) dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListConnectionsà la section Référence des AWS CLI commandes.

`list-hosts`

L'exemple de code suivant montre comment utiliser`list-hosts`.

AWS CLI

Pour répertorier les hôtes

L'`list-hosts`exemple suivant permet de récupérer la liste de tous les hôtes de votre compte.

```
aws codestar-connections list-hosts
```

Sortie :

```
{
    "Hosts": [
        {
            "Name": "My-Host",
            "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605",
            "ProviderType": "GitHubEnterpriseServer",
            "ProviderEndpoint": "https://my-instance.test.dev",
            "Status": "AVAILABLE"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section List hosts (CLI) dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListHostsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags

L'`list-tags-for-resource`exemple suivant extrait une liste de toutes les balises associées à la ressource de connexions spécifiée.

```
aws codestar-connections list-tags-for-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "ProjectA"
        },
        {
            "Key": "ReadOnly",
            "Value": "true"
        }
    ]
}
```

Pour plus d'informations, voir Afficher les balises d'une ressource de connexion dans le guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant associe un ensemble de balises fournies à une connexion. Utilisez cette commande pour ajouter ou modifier des balises.

```
aws codestar-connections tag-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \
    --tags Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Ajouter des balises à une ressource de connexion dans le Guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des AWS balises d'une ressource de connexion

Ce qui suit `untag-resource` supprime une balise de la ressource spécifiée.

```
aws codestar-connections untag-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \
    --tag-keys Project ReadOnly
```

Sortie :

```
{
    "Tags": []
}
```

Pour plus d'informations, voir Supprimer les balises d'une ressource de connexion dans le Guide de l'utilisateur de la console Developer Tools.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples d'Amazon Cognito Identity utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Cognito Identity.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-identity-pool`

L'exemple de code suivant montre comment utiliser`create-identity-pool`.

AWS CLI

Pour créer une réserve d’identités avec le fournisseur de réserves d’identités Cognito

Cet exemple crée un pool d'identités nommé MyIdentityPool. Il dispose d’un fournisseur de réserve d’identités Cognito. Les identités non authentifiées ne sont pas autorisées.

Commande :

```
aws cognito-identity create-identity-pool --identity-pool-
name MyIdentityPool --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateIdentityPoolà la section Référence des AWS CLI commandes.

`delete-identities`

L'exemple de code suivant montre comment utiliser`delete-identities`.

AWS CLI

Pour supprimer une réserve d’identités

Cet exemple supprime un pool d'identités.

Commande :

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Sortie :

```
{
  "UnprocessedIdentityIds": []
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteIdentitiesà la section Référence des AWS CLI commandes.

`delete-identity-pool`

L'exemple de code suivant montre comment utiliser`delete-identity-pool`.

AWS CLI

Pour supprimer une réserve d'identités

L'exemple `delete-identity-pool` suivant supprime la réserve d'identités spécifiée.

Commande :

```
aws cognito-identity delete-identity-pool \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteIdentityPoolà la section Référence des AWS CLI commandes.

`describe-identity-pool`

L'exemple de code suivant montre comment utiliser`describe-identity-pool`.

AWS CLI

Pour décrire un pool d'identités

Cet exemple décrit un pool d'identités.

Commande :

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeIdentityPoolà la section Référence des AWS CLI commandes.

`get-identity-pool-roles`

L'exemple de code suivant montre comment utiliser`get-identity-pool-roles`.

AWS CLI

Pour obtenir des rôles dans le pool d'identités

Cet exemple permet d'obtenir les rôles du pool d'identités.

Commande :

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
      "authenticated": "arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role",
      "unauthenticated": "arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolUnauth_Role"
```

```
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetIdentityPoolRolesà la section Référence des AWS CLI commandes.

`list-identity-pools`

L'exemple de code suivant montre comment utiliser`list-identity-pools`.

AWS CLI

Pour afficher les réserves d'identités

Cet exemple répertorie les réserves d'identités. Un maximum de 20 identités sont répertoriées.

Commande :

```
aws cognito-identity list-identity-pools --max-results 20
```

Sortie :

```
{
  "IdentityPools": [
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "MyIdentityPool"
      },
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "AnotherIdentityPool"
      },
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "IdentityPoolRegionA"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListIdentityPoolsà la section Référence des AWS CLI commandes.

`set-identity-pool-roles`

L'exemple de code suivant montre comment utiliser`set-identity-pool-roles`.

AWS CLI

Pour définir les rôles du pool d'identités

L'`set-identity-pool-roles`exemple suivant définit un rôle de pool d'identités.

```
aws cognito-identity set-identity-pool-roles \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \
    --roles authenticated="arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role"
```

- Pour plus de détails sur l'API, reportez-vous SetIdentityPoolRolesà la section Référence des AWS CLI commandes.

`update-identity-pool`

L'exemple de code suivant montre comment utiliser`update-identity-pool`.

AWS CLI

Pour mettre à jour un pool d'identités

Cet exemple met à jour un pool d'identités. Il définit le nom sur MyIdentityPool. Il ajoute Cognito en tant que fournisseur d'identité. Il interdit les identités non authentifiées.

Commande :

```
aws cognito-identity update-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111" --identity-pool-
name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
```

```
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateIdentityPoolà la section Référence des AWS CLI commandes.

Exemples d'utilisation du fournisseur d'identité Amazon Cognito AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du fournisseur AWS Command Line Interface d'identité Amazon Cognito.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-custom-attributes`

L'exemple de code suivant montre comment utiliser`add-custom-attributes`.

AWS CLI

Pour ajouter un attribut personnalisé

Cet exemple ajoute un attribut personnalisé CustomAttr 1 à un groupe d'utilisateurs. Il s'agit d'un type String qui nécessite un minimum de 1 caractère et un maximum de 15. Il n'est pas obligatoire.

Commande :

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaaa --custom-
attributes
 Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- Pour plus de détails sur l'API, reportez-vous AddCustomAttributesà la section Référence des AWS CLI commandes.

`admin-add-user-to-group`

L'exemple de code suivant montre comment utiliser`admin-add-user-to-group`.

AWS CLI

Pour ajouter un utilisateur à un groupe

Cet exemple ajoute l'utilisateur Jane au groupe MyGroup.

Commande :

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --
username Jane --group-name MyGroup
```

- Pour plus de détails sur l'API, reportez-vous AdminAddUserToGroupà la section Référence des AWS CLI commandes.

`admin-confirm-sign-up`

L'exemple de code suivant montre comment utiliser`admin-confirm-sign-up`.

AWS CLI

Pour confirmer l'enregistrement de l'utilisateur

Cet exemple confirme l'utilisateur jane@example.com.

Commande :

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- Pour plus de détails sur l'API, reportez-vous AdminConfirmSignUpà la section Référence des AWS CLI commandes.

`admin-create-user`

L'exemple de code suivant montre comment utiliser`admin-create-user`.

AWS CLI

Pour créer un utilisateur

L'`admin-create-user`exemple suivant crée un utilisateur avec l'adresse e-mail et le numéro de téléphone spécifiés dans les paramètres.

```
aws cognito-idp admin-create-user \
    --user-pool-id us-west-2_aaaaaaaaa \
    --username diego \
    --user-attributes Name=email,Value=diego@example.com
 Name=phone_number,Value="+15555551212" \
    --message-action SUPPRESS
```

Sortie :

```
{
    "User": {
        "Username": "diego",
        "Attributes": [
            {
                "Name": "sub",
                "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"
            },
            {
                "Name": "phone_number",
                "Value": "+15555551212"
            },
            {
                "Name": "email",
                "Value": "diego@example.com"
            }
        ],
        "UserCreateDate": 1548099495.428,
        "UserLastModifiedDate": 1548099495.428,
```

```
        "Enabled": true,
        "UserStatus": "FORCE_CHANGE_PASSWORD"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous AdminCreateUserà la section Référence des AWS CLI commandes.

`admin-delete-user-attributes`

L'exemple de code suivant montre comment utiliser`admin-delete-user-attributes`.

AWS CLI

Pour supprimer un attribut utilisateur

Cet exemple supprime un attribut personnalisé CustomAttr 1 pour l'utilisateur diego@example.com.

Commande :

```
aws cognito-idp admin-delete-user-attributes --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --user-attribute-names "custom:CustomAttr1"
```

- Pour plus de détails sur l'API, reportez-vous AdminDeleteUserAttributesà la section Référence des AWS CLI commandes.

`admin-delete-user`

L'exemple de code suivant montre comment utiliser`admin-delete-user`.

AWS CLI

Pour supprimer un utilisateur

Cet exemple supprime un utilisateur.

Commande :

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- Pour plus de détails sur l'API, reportez-vous AdminDeleteUserà la section Référence des AWS CLI commandes.

`admin-disable-provider-for-user`

L'exemple de code suivant montre comment utiliser`admin-disable-provider-for-user`.

AWS CLI

Pour dissocier un utilisateur fédéré d'un profil utilisateur local

L'`admin-disable-provider-for-user`exemple suivant déconnecte un utilisateur de Google de son profil local associé.

```
aws cognito-idp admin-disable-provider-for-user \
    --user-pool-id us-west-2_EXAMPLE \
    --
user ProviderAttributeName=Cognito_Subject,ProviderAttributeValue=0000000000000000,ProviderN
```

Pour plus d'informations, consultez la section Lier des utilisateurs fédérés à un profil utilisateur existant dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminDisableProviderForUserà la section Référence des AWS CLI commandes.

`admin-disable-user`

L'exemple de code suivant montre comment utiliser`admin-disable-user`.

AWS CLI

Pour empêcher un utilisateur de se connecter

L'`admin-disable-user`exemple suivant empêche l'utilisateur `diego@example.com` de se connecter.

```
aws cognito-idp admin-disable-user \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

Pour plus d'informations, consultez la section Gestion des utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminDisableUserà la section Référence des AWS CLI commandes.

`admin-enable-user`

L'exemple de code suivant montre comment utiliser`admin-enable-user`.

AWS CLI

Pour activer la connexion d'un utilisateur

L'`admin-enable-user`exemple suivant permet à l'utilisateur diego@example.com de se connecter.

```
aws cognito-idp admin-enable-user \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

Pour plus d'informations, consultez la section Gestion des utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminEnableUserà la section Référence des AWS CLI commandes.

`admin-forget-device`

L'exemple de code suivant montre comment utiliser`admin-forget-device`.

AWS CLI

Pour oublier un appareil

Cet exemple oublie le nom d'utilisateur de l'appareil jane@example.com

Commande :

```
aws cognito-idp admin-forget-device --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- Pour plus de détails sur l'API, reportez-vous AdminForgetDeviceà la section Référence des AWS CLI commandes.

`admin-get-device`

L'exemple de code suivant montre comment utiliser`admin-get-device`.

AWS CLI

Pour obtenir un appareil

L'`admin-get-device`exemple suivant montre un appareil pour l'utilisateur`diego`.

```
aws cognito-idp admin-get-device \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "Device": {
        "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeviceAttributes": [
            {
                "Name": "device_status",
                "Value": "valid"
            },
            {
                "Name": "device_name",
                "Value": "MyDevice"
            },
            {
                "Name": "dev:device_arn",
                "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/diego.us-
west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            {
                "Name": "dev:device_owner",
                "Value": "diego.us-west-2_EXAMPLE"
            },
            {
                "Name": "last_ip_used",
```

```
                "Value": "192.0.2.1"
            },
            {
                "Name": "dev:device_remembered_status",
                "Value": "remembered"
            },
            {
                "Name": "dev:device_sdk",
                "Value": "aws-sdk"
            }
        ],
        "DeviceCreateDate": 1715100742.022,
        "DeviceLastModifiedDate": 1723233651.167,
        "DeviceLastAuthenticatedDate": 1715100742.0
    }
}
```

Pour plus d'informations, consultez la section Utilisation des appareils utilisateur de votre groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminGetDeviceà la section Référence des AWS CLI commandes.

`admin-get-user`

L'exemple de code suivant montre comment utiliser`admin-get-user`.

AWS CLI

Pour obtenir un utilisateur

Cet exemple permet d'obtenir des informations sur le nom d'utilisateur jane@example.com.

Commande :

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

Sortie :

```
{
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",
  "Enabled": true,
```

```
  "UserStatus": "FORCE_CHANGE_PASSWORD",
  "UserCreateDate": 1548108509.537,
  "UserAttributes": [
      {
          "Name": "sub",
          "Value": "4320de44-2322-4620-999b-5e2e1c8df013"
      },
      {
          "Name": "email_verified",
          "Value": "true"
      },
      {
          "Name": "phone_number_verified",
          "Value": "true"
      },
      {
          "Name": "phone_number",
          "Value": "+01115551212"
      },
      {
          "Name": "email",
          "Value": "jane@example.com"
      }
  ],
  "UserLastModifiedDate": 1548108509.537
}
```

- Pour plus de détails sur l'API, reportez-vous AdminGetUserà la section Référence des AWS CLI commandes.

`admin-initiate-auth`

L'exemple de code suivant montre comment utiliser`admin-initiate-auth`.

AWS CLI

Pour connecter un utilisateur en tant qu'administrateur

L'`admin-initiate-auth`exemple suivant signe l'utilisateur diego@example.com. Cet exemple inclut également des métadonnées pour la protection contre les menaces et ClientMetadata pour les déclencheurs Lambda. L'utilisateur est configuré pour le TOTP MFA et est invité à fournir un code depuis son application d'authentification avant de pouvoir terminer l'authentification.

```
aws cognito-idp admin-initiate-auth \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --auth-flow ADMIN_USER_PASSWORD_AUTH \
    --auth-parameters USERNAME=diego@example.com,PASSWORD="My@Example
$Password3!",SECRET_HASH=ExampleEncodedClientIdSecretAndUsername= \
    --context-data="{\"EncodedData\":\"abc123example\",\"HttpHeaders\":
[{\"headerName\":\"UserAgent\",\"headerValue\":\"Mozilla/5.0 (Windows NT 6.1;
 Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0\"}],\"IpAddress\":\"192.0.2.1\",
\"ServerName\":\"example.com\",\"ServerPath\":\"/login\"}" \
    --client-metadata="{\"MyExampleKey\": \"MyExampleValue\"}"
```

Sortie :

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeExample...",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "MyAuthenticatorApp",
        "USER_ID_FOR_SRP": "diego@example.com"
    }
}
```

Pour plus d'informations, consultez le flux d'authentification des administrateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminInitiateAuthà la section Référence des AWS CLI commandes.

`admin-link-provider-for-user`

L'exemple de code suivant montre comment utiliser`admin-link-provider-for-user`.

AWS CLI

Pour lier un utilisateur local à un utilisateur fédéré

L'`admin-link-provider-for-user`exemple suivant lie l'utilisateur local diego à un utilisateur qui effectuera une connexion fédérée avec Google.

```
aws cognito-idp admin-link-provider-for-user \
```

```
    --user-pool-id us-west-2_EXAMPLE \
    --destination-user ProviderName=Cognito,ProviderAttributeValue=diego \
    --source-
user ProviderAttributeName=Cognito_Subject,ProviderAttributeValue=0000000000000000,ProviderN
```

Pour plus d'informations, consultez la section Lier des utilisateurs fédérés à un profil utilisateur existant dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminLinkProviderForUserà la section Référence des AWS CLI commandes.

`admin-list-devices`

L'exemple de code suivant montre comment utiliser`admin-list-devices`.

AWS CLI

Pour répertorier les appareils d'un utilisateur

L'`admin-list-devices`exemple suivant répertorie les appareils de l'utilisateur diego.

```
aws cognito-idp admin-list-devices \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego \
    --limit 1
```

Sortie :

```
{
    "Devices": [
        {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "device_name",
                    "Value": "MyDevice"
                },
                {
```

```
                    "Name": "dev:device_arn",
                    "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/
diego.us-west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                {
                    "Name": "dev:device_owner",
                    "Value": "diego.us-west-2_EXAMPLE"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.1"
                },
                {
                    "Name": "dev:device_remembered_status",
                    "Value": "remembered"
                },
                {
                    "Name": "dev:device_sdk",
                    "Value": "aws-sdk"
                }
            ],
            "DeviceCreateDate": 1715100742.022,
            "DeviceLastModifiedDate": 1723233651.167,
            "DeviceLastAuthenticatedDate": 1715100742.0
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des appareils utilisateur de votre groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminListDevicesà la section Référence des AWS CLI commandes.

`admin-list-groups-for-user`

L'exemple de code suivant montre comment utiliser`admin-list-groups-for-user`.

AWS CLI

Pour répertorier les groupes d'un utilisateur

Cet exemple répertorie les groupes pour le nom d'utilisateur jane@example.com.

Commande :

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

Sortie :

```
{
  "Groups": [
      {
          "Description": "Sample group",
          "Precedence": 1,
          "LastModifiedDate": 1548097827.125,
          "RoleArn": "arn:aws:iam::111111111111:role/SampleRole",
          "GroupName": "SampleGroup",
          "UserPoolId": "us-west-2_aaaaaaaaa",
          "CreationDate": 1548097827.125
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous AdminListGroupsForUserà la section Référence des AWS CLI commandes.

`admin-list-user-auth-events`

L'exemple de code suivant montre comment utiliser`admin-list-user-auth-events`.

AWS CLI

Pour répertorier les événements d'autorisation pour un utilisateur

L'`admin-list-user-auth-events`exemple suivant répertorie le dernier événement du journal d'activité utilisateur pour l'utilisateur diego.

```
aws cognito-idp admin-list-user-auth-events \
    --user-pool-id us-west-2_ywDJHlIfU \
    --username brcotter+050123 \
    --max-results 1
```

Sortie :

```
{
    "AuthEvents": [
        {
            "EventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "EventType": "SignIn",
            "CreationDate": 1726694203.495,
            "EventResponse": "InProgress",
            "EventRisk": {
                "RiskDecision": "AccountTakeover",
                "RiskLevel": "Medium",
                "CompromisedCredentialsDetected": false
            },
            "ChallengeResponses": [
                {
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
            "EventContextData": {
                "IpAddress": "192.0.2.1",
                "City": "Seattle",
                "Country": "United States"
            }
        }
    ],
    "NextToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222#2024-09-18T21:16:43.495Z"
}
```

Pour plus d'informations, consultez la section Affichage et exportation de l'historique des événements utilisateur dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminListUserAuthEventsà la section Référence des AWS CLI commandes.

`admin-remove-user-from-group`

L'exemple de code suivant montre comment utiliser`admin-remove-user-from-group`.

AWS CLI

Pour supprimer un utilisateur d'un groupe

Cet exemple supprime jane@example.com de SampleGroup.

Commande :

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com --group-name SampleGroup
```

- Pour plus de détails sur l'API, reportez-vous AdminRemoveUserFromGroupà la section Référence des AWS CLI commandes.

`admin-reset-user-password`

L'exemple de code suivant montre comment utiliser`admin-reset-user-password`.

AWS CLI

Pour réinitialiser le mot de passe d'un utilisateur

Cet exemple réinitialise le mot de passe pour diego@example.com.

Commande :

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- Pour plus de détails sur l'API, reportez-vous AdminResetUserPasswordà la section Référence des AWS CLI commandes.

`admin-respond-to-auth-challenge`

L'exemple de code suivant montre comment utiliser`admin-respond-to-auth-challenge`.

AWS CLI

Pour répondre à un défi d'authentification

Il existe de nombreuses manières de répondre aux différents défis d'authentification, en fonction de votre flux d'authentification, de la configuration du groupe d'utilisateurs et des paramètres utilisateur. L'`admin-respond-to-auth-challenge`exemple suivant fournit un code MFA TOTP pour diego@example.com et termine la connexion. La mémorisation des appareils de ce groupe d'utilisateurs est activée, de sorte que le résultat de l'authentification renvoie également une nouvelle clé de périphérique.

```
aws cognito-idp admin-respond-to-auth-challenge \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=diego@example.com,SOFTWARE_TOKEN_MFA_CODE=000000
 \
    --session AYABeExample...
```

Sortie :

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-ExAmPlE1"
        }
    }
}
```

Pour plus d'informations, consultez le flux d'authentification des administrateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminRespondToAuthChallengeà la section Référence des AWS CLI commandes.

`admin-set-user-mfa-preference`

L'exemple de code suivant montre comment utiliser`admin-set-user-mfa-preference`.

AWS CLI

Pour définir les préférences MFA de l'utilisateur

Cet exemple définit la préférence MFA SMS pour le nom d'utilisateur diego@example.com.

Commande :

```
aws cognito-idp admin-set-user-mfa-preference --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --sms-mfa-settings Enabled=false,PreferredMfa=false
```

- Pour plus de détails sur l'API, reportez-vous AdminSetUserMfaPreferenceà la section Référence des AWS CLI commandes.

`admin-set-user-password`

L'exemple de code suivant montre comment utiliser`admin-set-user-password`.

AWS CLI

Pour définir un mot de passe utilisateur en tant qu'administrateur

L'`admin-set-user-password`exemple suivant définit définitivement le mot de passe pour diego@example.com.

```
aws cognito-idp admin-set-user-password \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com \
    --password MyExamplePassword1! \
    --permanent
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Mots de passe, récupération des mots de passe et politiques relatives aux mots de passe dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminSetUserPasswordà la section Référence des AWS CLI commandes.

`admin-set-user-settings`

L'exemple de code suivant montre comment utiliser`admin-set-user-settings`.

AWS CLI

Pour définir les paramètres utilisateur

Cet exemple définit la préférence de livraison MFA pour le nom d'utilisateur diego@example.com sur EMAIL.

Commande :

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- Pour plus de détails sur l'API, reportez-vous AdminSetUserSettingsà la section Référence des AWS CLI commandes.

`admin-update-auth-event-feedback`

L'exemple de code suivant montre comment utiliser`admin-update-auth-event-feedback`.

AWS CLI

Pour fournir des commentaires concernant un événement d'autorisation

Cet exemple définit la valeur de feedback pour un événement d'autorisation identifié par event-id sur Valid.

Commande :

```
aws cognito-idp admin-update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa
 --username diego@example.com --event-id c2c2cf89-c0d3-482d-aba6-99d78a5b0bfe --
feedback-value Valid
```

- Pour plus de détails sur l'API, reportez-vous AdminUpdateAuthEventFeedbackà la section Référence des AWS CLI commandes.

`admin-update-device-status`

L'exemple de code suivant montre comment utiliser`admin-update-device-status`.

AWS CLI

Pour mettre à jour l'état de l'appareil

Cet exemple définit l'état mémorisé du périphérique identifié par device-key sur not_remembered.

Commande :

```
aws cognito-idp admin-update-device-status --user-pool-id us-west-2_aaaaaaaaa
 --username diego@example.com --device-key xxxx  --device-remembered-
status not_remembered
```

- Pour plus de détails sur l'API, reportez-vous AdminUpdateDeviceStatusà la section Référence des AWS CLI commandes.

`admin-update-user-attributes`

L'exemple de code suivant montre comment utiliser`admin-update-user-attributes`.

AWS CLI

Pour mettre à jour les attributs utilisateur

Cet exemple met à jour un attribut utilisateur personnalisé CustomAttr 1 pour l'utilisateur diego@example.com.

Commande :

```
aws cognito-idp admin-update-user-attributes --user-pool-id us-
west-2_aaaaaaaaa --username diego@example.com --user-attributes
 Name="custom:CustomAttr1",Value="Purple"
```

- Pour plus de détails sur l'API, reportez-vous AdminUpdateUserAttributesà la section Référence des AWS CLI commandes.

`admin-user-global-sign-out`

L'exemple de code suivant montre comment utiliser`admin-user-global-sign-out`.

AWS CLI

Pour déconnecter un utilisateur en tant qu'administrateur

L'`admin-user-global-sign-out`exemple suivant déconnecte l'utilisateur diego@example.com.

```
aws cognito-idp admin-user-global-sign-out \
```

```
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

Pour plus d'informations, consultez la section Authentification auprès d'un groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AdminUserGlobalSignOutà la section Référence des AWS CLI commandes.

`associate-software-token`

L'exemple de code suivant montre comment utiliser`associate-software-token`.

AWS CLI

Pour générer une clé secrète pour une application d'authentification MFA

L'`associate-software-token`exemple suivant génère une clé privée TOTP pour un utilisateur qui s'est connecté et a reçu un jeton d'accès. La clé privée qui en résulte peut être saisie manuellement dans une application d'authentification, ou les applications peuvent la restituer sous forme de code QR que l'utilisateur peut scanner.

```
aws cognito-idp associate-software-token \
    --access-token eyJra456defEXAMPLE
```

Sortie :

```
{
    "SecretCode": "QWERTYUIOP123456EXAMPLE"
}
```

Pour plus d'informations, consultez la section MFA du jeton logiciel TOTP dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateSoftwareTokenà la section Référence des AWS CLI commandes.

`change-password`

L'exemple de code suivant montre comment utiliser`change-password`.

AWS CLI

Pour modifier un mot de passe

Cet exemple modifie un mot de passe.

Commande :

```
aws cognito-idp change-password --previous-password OldPassword --proposed-
password NewPassword --access-token ACCESS_TOKEN
```

- Pour plus de détails sur l'API, reportez-vous ChangePasswordà la section Référence des AWS CLI commandes.

`confirm-device`

L'exemple de code suivant montre comment utiliser`confirm-device`.

AWS CLI

Pour confirmer l'appareil d'un utilisateur

L'`confirm-device`exemple suivant ajoute un nouvel appareil mémorisé pour l'utilisateur actuel.

```
aws cognito-idp confirm-device \
   --access-token eyJra456defEXAMPLE \
   --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
   --device-secret-verifier-
config PasswordVerifier=TXlWZXJpZmllclN0cmluZw,Salt=TXlTUlBTYWx0
```

Sortie :

```
{
     "UserConfirmationNecessary": false
}
```

Pour plus d'informations, consultez la section Utilisation des appareils utilisateur de votre groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ConfirmDeviceà la section Référence des AWS CLI commandes.

`confirm-forgot-password`

L'exemple de code suivant montre comment utiliser`confirm-forgot-password`.

AWS CLI

Pour confirmer un mot de passe oublié

Cet exemple confirme un mot de passe oublié pour le nom d'utilisateur diego@example.com.

Commande :

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Pour plus de détails sur l'API, reportez-vous ConfirmForgotPasswordà la section Référence des AWS CLI commandes.

`confirm-sign-up`

L'exemple de code suivant montre comment utiliser`confirm-sign-up`.

AWS CLI

Pour confirmer l’inscription

Cet exemple confirme l’inscription pour le nom d’utilisateur diego@example.com.

Commande :

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- Pour plus de détails sur l'API, reportez-vous ConfirmSignUpà la section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour créer un groupe

Cet exemple crée un groupe avec une description.

Commande :

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroup --description "New group."
```

Sortie :

```
{
  "Group": {
      "GroupName": "MyNewGroup",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New group.",
      "LastModifiedDate": 1548270073.795,
      "CreationDate": 1548270073.795
  }
}
```

Pour créer un groupe doté d'un rôle et d'une priorité

Cet exemple crée un groupe avec une description. Il inclut également un rôle et une priorité.

Commande :

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroupWithRole --description "New group with a role." --role-
arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

Sortie :

```
{
  "Group": {
      "GroupName": "MyNewGroupWithRole",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New group with a role.",
      "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
      "Precedence": 2,
```

```
        "LastModifiedDate": 1548270211.761,
        "CreationDate": 1548270211.761
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`create-identity-provider`

L'exemple de code suivant montre comment utiliser`create-identity-provider`.

AWS CLI

Exemple 1 : pour créer un fournisseur d'identité SAML (IdP) de groupe d'utilisateurs avec une URL de métadonnées

L'`create-identity-provider`exemple suivant crée un nouvel IdP SAML avec des métadonnées provenant d'une URL publique, un mappage d'attributs et deux identifiants.

```
aws cognito-idp create-identity-provider \
    --user-pool-id us-west-2_EXAMPLE \
    --provider-name MySAML \
    --provider-type SAML \
    --provider-
details IDPInit=true,IDPSignout=true,EncryptedResponses=true,MetadataURL=https://
auth.example.com/sso/saml/metadata,RequestSigningAlgorithm=rsa-sha256 \
    --attribute-mapping email=emailaddress,phone_number=phone,custom:111=department
 \
    --idp-identifiers CorpSAML WestSAML
```

Sortie :

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MySAML",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate": "MIICvTCCAaEXAMPLE",
            "EncryptedResponses": "true",
            "IDPInit": "true",
```

```
            "IDPSignout": "true",
            "MetadataURL": "https://auth.example.com/sso/saml/metadata",
            "RequestSigningAlgorithm": "rsa-sha256",
            "SLORedirectBindingURI": "https://auth.example.com/slo/saml",
            "SSORedirectBindingURI": "https://auth.example.com/sso/saml"
        },
        "AttributeMapping": {
            "custom:111": "department",
            "emailaddress": "email",
            "phone": "phone_number"
        },
        "IdpIdentifiers": [
            "CorpSAML",
            "WestSAML"
        ],
        "LastModifiedDate": 1726853833.977,
        "CreationDate": 1726853833.977
    }
}
```

Pour plus d'informations, consultez la section Ajouter la connexion à un groupe d'utilisateurs via un tiers dans le manuel Amazon Cognito Developer Guide.

Exemple 2 : pour créer un fournisseur d'identité SAML (IdP) de groupe d'utilisateurs avec un fichier de métadonnées

L'`create-identity-provider`exemple suivant crée un nouvel IdP SAML avec les métadonnées d'un fichier, un mappage d'attributs et deux identifiants. La syntaxe du fichier peut varier d'un système d'exploitation à l'autre dans le `--provider-details` paramètre. Il est plus simple de créer un fichier d'entrée JSON pour cette opération. :

```
aws cognito-idp create-identity-provider \
    --cli-input-json file://.\SAML-identity-provider.json
```

Contenu de `SAML-identity-provider.json` :

```
{
    "AttributeMapping": {
        "email" : "idp_email",
        "email_verified" : "idp_email_verified"
    },
    "IdpIdentifiers": [ "platform" ],
```

```
    "ProviderDetails": {
        "MetadataFile": "<md:EntityDescriptor xmlns:md=
\"urn:oasis:names:tc:SAML:2.0:metadata\" entityID=\"http://www.example.com/
sso\"><md:IDPSSODescriptor WantAuthnRequestsSigned=\"false\"
 protocolSupportEnumeration=\"urn:oasis:names:tc:SAML:2.0:protocol
\"><md:KeyDescriptor use=\"signing\"><ds:KeyInfo xmlns:ds=\"http://
www.w3.org/2000/09/xmldsig#
\"><ds:X509Data><ds:X509Certificate>[IDP_CERTIFICATE_DATA]</ds:X509Certificate></
ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/slo/saml\"/><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect\" Location=\"https://www.example.com/slo/saml\"/
><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/sso/saml\"/><md:SingleSignOnService Binding=
\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect\" Location=\"https://
www.example.com/sso/saml\"/></md:IDPSSODescriptor></md:EntityDescriptor>",
        "IDPSignout" : "true",
        "RequestSigningAlgorithm" : "rsa-sha256",
        "EncryptedResponses" : "true",
        "IDPInit" : "true"
    },
    "ProviderName": "MySAML2",
    "ProviderType": "SAML",
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

Sortie :

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MySAML2",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate":
 "[USER_POOL_ENCRYPTION_CERTIFICATE_DATA]",
            "EncryptedResponses": "true",
            "IDPInit": "true",
            "IDPSignout": "true",
```

```
            "MetadataFile": "<md:EntityDescriptor xmlns:md=
\"urn:oasis:names:tc:SAML:2.0:metadata\" entityID=\"http://www.example.com/
sso\"><md:IDPSSODescriptor WantAuthnRequestsSigned=\"false\"
 protocolSupportEnumeration=\"urn:oasis:names:tc:SAML:2.0:protocol
\"><md:KeyDescriptor use=\"signing\"><ds:KeyInfo xmlns:ds=\"http://
www.w3.org/2000/09/xmldsig#
\"><ds:X509Data><ds:X509Certificate>[IDP_CERTIFICATE_DATA]</ds:X509Certificate></
ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/slo/saml\"/><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect\" Location=\"https://www.example.com/slo/saml\"/
><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/sso/saml\"/><md:SingleSignOnService Binding=
\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect\" Location=\"https://
www.example.com/sso/saml\"/></md:IDPSSODescriptor></md:EntityDescriptor>",
            "RequestSigningAlgorithm": "rsa-sha256",
            "SLORedirectBindingURI": "https://www.example.com/slo/saml",
            "SSORedirectBindingURI": "https://www.example.com/sso/saml"
        },
        "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified"
        },
        "IdpIdentifiers": [
            "platform"
        ],
        "LastModifiedDate": 1726855290.731,
        "CreationDate": 1726855290.731
    }
}
```

Pour plus d'informations, consultez la section Ajouter la connexion à un groupe d'utilisateurs via un tiers dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateIdentityProviderà la section Référence des AWS CLI commandes.

`create-resource-server`

L'exemple de code suivant montre comment utiliser`create-resource-server`.

AWS CLI

Pour créer un client de groupe d'utilisateurs

L'`create-resource-server`exemple suivant crée un nouveau serveur de ressources avec des étendues personnalisées.

```
aws cognito-idp create-resource-server \
    --user-pool-id us-west-2_EXAMPLE \
    --identifier solar-system-data \
    --name "Solar system object tracker" \
    --scopes ScopeName=sunproximity.read,ScopeDescription="Distance in AU from Sol"
 ScopeName=asteroids.add,ScopeDescription="Enter a new asteroid"
```

Sortie :

```
{
    "ResourceServer": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "Identifier": "solar-system-data",
        "Name": "Solar system object tracker",
        "Scopes": [
            {
                "ScopeName": "sunproximity.read",
                "ScopeDescription": "Distance in AU from Sol"
            },
            {
                "ScopeName": "asteroids.add",
                "ScopeDescription": "Enter a new asteroid"
            }
        ]
    }
}
```

Pour plus d'informations, consultez les sections Scopes, M2M et APIs avec serveurs de ressources dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateResourceServerà la section Référence des AWS CLI commandes.

`create-user-import-job`

L'exemple de code suivant montre comment utiliser`create-user-import-job`.

AWS CLI

Pour créer une tâche d'importation utilisateur

Cet exemple crée une tâche d'importation utilisateur nommée MyImportJob.

Pour plus d'informations sur l'importation d'utilisateurs, voir Importation d'utilisateurs dans des groupes d'utilisateurs à partir d'un fichier CSV.

Commande :

```
aws cognito-idp create-user-import-job --user-pool-id us-west-2_aaaaaaaaa --
job-name MyImportJob --cloud-watch-logs-role-arn arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole
```

Sortie :

```
{
  "UserImportJob": {
      "JobName": "MyImportJob",
      "JobId": "import-qQ0DCt2fRh",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "PreSignedUrl": "PRE_SIGNED_URL",
      "CreationDate": 1548271795.471,
      "Status": "Created",
      "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
      "ImportedUsers": 0,
      "SkippedUsers": 0,
      "FailedUsers": 0
  }
}
```

Téléchargez le fichier .csv avec curl à l'aide de l'URL pré-signée :

Commande :

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

- Pour plus de détails sur l'API, reportez-vous CreateUserImportJobà la section Référence des AWS CLI commandes.

`create-user-pool-client`

L'exemple de code suivant montre comment utiliser`create-user-pool-client`.

AWS CLI

Pour créer un client de groupe d'utilisateurs

L'`create-user-pool-client`exemple suivant crée un nouveau client de groupe d'utilisateurs avec un secret client, des attributs de lecture et d'écriture explicites, une connexion avec un nom d'utilisateur-mot de passe et des flux SRP, une connexion avec trois, un accès à un sous-ensemble de champs d'application IdPs, des PinPoint analyses et une durée de OAuth validité de session d'authentification étendue.

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name MyTestClient \
    --generate-secret \
    --refresh-token-validity 10 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days \
    --read-attributes email phone_number email_verified phone_number_verified \
    --write-attributes email phone_number \
    --explicit-auth-
flows ALLOW_USER_PASSWORD_AUTH ALLOW_USER_SRP_AUTH ALLOW_REFRESH_TOKEN_AUTH \
    --supported-identity-providers Google Facebook MyOIDC \
    --callback-urls https://www.amazon.com https://example.com http://
localhost:8001 myapp://example \
    --allowed-o-auth-flows code implicit \
    --allowed-o-auth-scopes openid profile aws.cognito.signin.user.admin solar-
system-data/asteroids.add \
    --allowed-o-auth-flows-user-pool-client \
    --analytics-configuration ApplicationArn=arn:aws:mobiletargeting:us-
west-2:767671399759:apps/thisisanexamplepinpointapplicationid,UserDataShared=TRUE \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --enable-propagate-additional-user-context-data \
    --auth-session-validity 4
```

Sortie :


```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "MyTestClient",
        "ClientId": "123abc456defEXAMPLE",
        "ClientSecret": "this1234is5678my91011example1213client1415secret",
        "LastModifiedDate": 1726788459.464,
        "CreationDate": 1726788459.464,
        "RefreshTokenValidity": 10,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "email_verified",
            "phone_number_verified",
            "phone_number",
            "email"
        ],
        "WriteAttributes": [
            "phone_number",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "MyOIDC",
            "Facebook"
        ],
        "CallbackURLs": [
            "https://example.com",
            "https://www.amazon.com",
            "myapp://example",
            "http://localhost:8001"
        ],
```

```
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile",
            "solar-system-data/asteroids.add"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "AnalyticsConfiguration": {
            "ApplicationArn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/
thisisanexamplepinpointapplicationid",
            "RoleArn": "arn:aws:iam::123456789012:role/aws-service-role/cognito-
idp.amazonaws.com/AWSServiceRoleForAmazonCognitoIdp",
            "UserDataShared": true
        },
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": true,
        "AuthSessionValidity": 4
    }
}
```

Pour plus d'informations, consultez la section Paramètres spécifiques à l'application avec les clients d'applications dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateUserPoolClientà la section Référence des AWS CLI commandes.

`create-user-pool-domain`

L'exemple de code suivant montre comment utiliser`create-user-pool-domain`.

AWS CLI

Exemple 1 : pour créer un domaine de groupe d'utilisateurs

L'`create-user-pool-domain`exemple suivant crée un nouveau domaine personnalisé.

```
aws cognito-idp create-user-pool-domain \
    --user-pool-id us-west-2_EXAMPLE \
```

```
    --domain auth.example.com \
    --custom-domain-config CertificateArn=arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Sortie :

```
{
    "CloudFrontDomain": "example1domain.cloudfront.net"
}
```

Pour plus d'informations, consultez la section Configuration d'un domaine de groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

Exemple 2 : pour créer un domaine de groupe d'utilisateurs

L'`create-user-pool-domain`exemple suivant crée un nouveau domaine avec un préfixe appartenant au service.

```
aws cognito-idp create-user-pool-domain \
    --user-pool-id us-west-2_EXAMPLE2 \
    --domain mydomainprefix
```

Pour plus d'informations, consultez la section Configuration d'un domaine de groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateUserPoolDomainà la section Référence des AWS CLI commandes.

`create-user-pool`

L'exemple de code suivant montre comment utiliser`create-user-pool`.

AWS CLI

Pour créer un groupe d’utilisateurs configuré de manière minimale

Cet exemple crée un groupe d'utilisateurs nommé à MyUserPool l'aide des valeurs par défaut. Il n’y a aucun attribut obligatoire ni aucun client d’application. La MFA et la sécurité avancée sont désactivés.

Commande :

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Sortie :

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "given_name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "family_name",
              "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
```

```
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
```

```
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
            "Mutable": true
        }
    ],
```

```
        "MfaConfiguration": "OFF",
        "Name": "MyUserPool",
        "LastModifiedDate": 1547833345.777,
        "AdminCreateUserConfig": {
            "UnusedAccountValidityDays": 7,
            "AllowAdminCreateUserOnly": false
        },
        "EmailConfiguration": {},
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "CreationDate": 1547833345.777,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

Pour créer un groupe d'utilisateurs avec deux attributs requis

Cet exemple crée un groupe d'utilisateurs MyUserPool. Le groupe est configuré pour accepter l'e-mail en tant qu'attribut de nom d'utilisateur. Il définit également l'adresse e-mail source sur une adresse validée à l'aide d'Amazon Simple Email Service.

Commande :

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-
attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-
east-1:111111111111:identity/
jane@example.com",ReplyToEmailAddress="jane@example.com"
```

Sortie :

```
{
    "UserPool": {
        "SchemaAttributes": [
            {
```

```
            "Name": "sub",
            "StringAttributeConstraints": {
                "MinLength": "1",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": true,
            "AttributeDataType": "String",
            "Mutable": false
        },
        {
            "Name": "name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "given_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "family_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
```

```
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
```

```
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
```

```
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
```

```
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
            "Mutable": true
        }
    ],
    "MfaConfiguration": "OFF",
    "Name": "MyUserPool",
    "LastModifiedDate": 1547837788.189,
    "AdminCreateUserConfig": {
        "UnusedAccountValidityDays": 7,
        "AllowAdminCreateUserOnly": false
    },
    "EmailConfiguration": {
        "ReplyToEmailAddress": "jane@example.com",
```

```
            "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
        },
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "UsernameAttributes": [
            "email"
        ],
        "CreationDate": 1547837788.189,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateUserPoolà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe

Cet exemple supprime un groupe.

Commande :

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaaa  --group-
name MyGroupName
```

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`delete-identity-provider`

L'exemple de code suivant montre comment utiliser`delete-identity-provider`.

AWS CLI

Pour supprimer un fournisseur d'identité

Cet exemple supprime un fournisseur d'identité.

Commande :

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

- Pour plus de détails sur l'API, reportez-vous DeleteIdentityProviderà la section Référence des AWS CLI commandes.

`delete-resource-server`

L'exemple de code suivant montre comment utiliser`delete-resource-server`.

AWS CLI

Pour supprimer un serveur de ressources

Cet exemple supprime un serveur de ressources nommé weather.example.com.

Commande :

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

- Pour plus de détails sur l'API, reportez-vous DeleteResourceServerà la section Référence des AWS CLI commandes.

`delete-user-attributes`

L'exemple de code suivant montre comment utiliser`delete-user-attributes`.

AWS CLI

Pour supprimer un attribut utilisateur

L'`delete-user-attributes`exemple suivant supprime l'attribut personnalisé « custom:attribute » de l'utilisateur actuellement connecté.

```
aws cognito-idp delete-user-attributes \
    --access-token ACCESS_TOKEN \
    --user-attribute-names "custom:department"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des attributs utilisateur dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteUserAttributesà la section Référence des AWS CLI commandes.

`delete-user-pool-client`

L'exemple de code suivant montre comment utiliser`delete-user-pool-client`.

AWS CLI

Pour supprimer un client du groupe d'utilisateurs

Cet exemple supprime un client du groupe d'utilisateurs.

Commande :

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-
id 38fjsnc484p94kpqsnet7mpld0
```

- Pour plus de détails sur l'API, reportez-vous DeleteUserPoolClientà la section Référence des AWS CLI commandes.

`delete-user-pool-domain`

L'exemple de code suivant montre comment utiliser`delete-user-pool-domain`.

AWS CLI

Pour supprimer le domaine d'un groupe d'utilisateurs

L'`delete-user-pool-domain`exemple suivant supprime un domaine de groupe d'utilisateurs nommé `my-domain`

```
aws cognito-idp delete-user-pool-domain \
    --user-pool-id us-west-2_aaaaaaaaa \
    --domain my-domain
```

- Pour plus de détails sur l'API, reportez-vous DeleteUserPoolDomainà la section Référence des AWS CLI commandes.

`delete-user-pool`

L'exemple de code suivant montre comment utiliser`delete-user-pool`.

AWS CLI

Pour supprimer un groupe d'utilisateurs

Cet exemple supprime un groupe d'utilisateurs à l'aide de l'identifiant du groupe d'utilisateurs us-west-2_aaaaaaaaa.

Commande :

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- Pour plus de détails sur l'API, reportez-vous DeleteUserPoolà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

Cet exemple supprime un utilisateur.

Commande :

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`describe-identity-provider`

L'exemple de code suivant montre comment utiliser`describe-identity-provider`.

AWS CLI

Pour décrire un fournisseur d'identité

Cet exemple décrit un fournisseur d'identité nommé Facebook.

Commande :

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

Sortie :

```
{
  "IdentityProvider": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ProviderName": "Facebook",
      "ProviderType": "Facebook",
      "ProviderDetails": {
          "attributes_url": "https://graph.facebook.com/me?fields=",
          "attributes_url_add_attributes": "true",
          "authorize_scopes": myscope",
          "authorize_url": "https://www.facebook.com/v2.9/dialog/oauth",
          "client_id": "11111",
          "client_secret": "11111",
          "token_request_method": "GET",
          "token_url": "https://graph.facebook.com/v2.9/oauth/access_token"
      },
      "AttributeMapping": {
          "username": "id"
      },
```

```
        "IdpIdentifiers": [],
        "LastModifiedDate": 1548105901.736,
        "CreationDate": 1548105901.736
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeIdentityProviderà la section Référence des AWS CLI commandes.

`describe-resource-server`

L'exemple de code suivant montre comment utiliser`describe-resource-server`.

AWS CLI

Pour décrire un serveur de ressources

Cet exemple décrit le serveur de ressources weather.example.com.

Commande :

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

Sortie :

```
{
    "ResourceServer": {
        "UserPoolId": "us-west-2_aaaaaaaaa",
        "Identifier": "weather.example.com",
        "Name": "Weather",
        "Scopes": [
            {
                "ScopeName": "weather.update",
                "ScopeDescription": "Update weather forecast"
            },
            {
                "ScopeName": "weather.read",
                "ScopeDescription": "Read weather forecasts"
            },
            {
                "ScopeName": "weather.delete",
                "ScopeDescription": "Delete a weather forecast"
```

```
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeResourceServerà la section Référence des AWS CLI commandes.

`describe-risk-configuration`

L'exemple de code suivant montre comment utiliser`describe-risk-configuration`.

AWS CLI

Pour décrire une configuration de risque

Cet exemple décrit la configuration des risques associée au pool us-west-2_aaaaaaaaa.

Commande :

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaa
```

Sortie :

```
{
  "RiskConfiguration": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "CompromisedCredentialsRiskConfiguration": {
          "EventFilter": [
              "SIGN_IN",
              "SIGN_UP",
              "PASSWORD_CHANGE"
          ],
          "Actions": {
              "EventAction": "BLOCK"
          }
      },
      "AccountTakeoverRiskConfiguration": {
          "NotifyConfiguration": {
              "From": "diego@example.com",
              "ReplyTo": "diego@example.com",
              "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
diego@example.com",
```

```
            "BlockEmail": {
                "Subject": "Blocked sign-in attempt",
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "TextBody": "We blocked an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
            },
            "NoActionEmail": {
                "Subject": "New sign-in attempt",
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
            },
            "MfaEmail": {
                "Subject": "New sign-in attempt",
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
 context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
 you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "TextBody": "We required you to use multi-factor authentication
 for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-
```

```
name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should
 change your password and notify us by clicking on {one-click-link-invalid}\nIf this
 sign-in was by you, you can follow {one-click-link-valid} to let us know"
                }
            },
            "Actions": {
                "LowAction": {
                    "Notify": true,
                    "EventAction": "NO_ACTION"
                },
                "MediumAction": {
                    "Notify": true,
                    "EventAction": "MFA_IF_CONFIGURED"
                },
                "HighAction": {
                    "Notify": true,
                    "EventAction": "MFA_IF_CONFIGURED"
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeRiskConfigurationà la section Référence des AWS CLI commandes.

`describe-user-import-job`

L'exemple de code suivant montre comment utiliser`describe-user-import-job`.

AWS CLI

Pour décrire une tâche d'importation par un utilisateur

Cet exemple décrit une tâche de saisie utilisateur.

Pour plus d'informations sur l'importation d'utilisateurs, voir Importation d'utilisateurs dans des groupes d'utilisateurs à partir d'un fichier CSV.

Commande :

```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-
id import-TZqNQvDRnW
```

Sortie :

```
{
  "UserImportJob": {
      "JobName": "import-Test1",
      "JobId": "import-TZqNQvDRnW",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "PreSignedUrl": "PRE_SIGNED URL",
      "CreationDate": 1548271708.512,
      "Status": "Created",
      "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
      "ImportedUsers": 0,
      "SkippedUsers": 0,
      "FailedUsers": 0
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeUserImportJobà la section Référence des AWS CLI commandes.

`describe-user-pool-client`

L'exemple de code suivant montre comment utiliser`describe-user-pool-client`.

AWS CLI

Pour décrire un client de groupe d'utilisateurs

Cet exemple décrit un client de groupe d'utilisateurs.

Commande :

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 38fjsnc484p94kpqsnet7mpld0
```

Sortie :

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyApp",
```

```
        "ClientId": "38fjsnc484p94kpqsnet7mpld0",
        "ClientSecret": "CLIENT_SECRET",
        "LastModifiedDate": 1548108676.163,
        "CreationDate": 1548108676.163,
        "RefreshTokenValidity": 30,
        "ReadAttributes": [
            "address",
            "birthdate",
            "custom:CustomAttr1",
            "custom:CustomAttr2",
            "email",
            "email_verified",
            "family_name",
            "gender",
            "given_name",
            "locale",
            "middle_name",
            "name",
            "nickname",
            "phone_number",
            "phone_number_verified",
            "picture",
            "preferred_username",
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "WriteAttributes": [
            "address",
            "birthdate",
            "custom:CustomAttr1",
            "custom:CustomAttr2",
            "email",
            "family_name",
            "gender",
            "given_name",
            "locale",
            "middle_name",
            "name",
            "nickname",
            "phone_number",
            "picture",
            "preferred_username",
```

```
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "ExplicitAuthFlows": [
            "ADMIN_NO_SRP_AUTH",
            "USER_PASSWORD_AUTH"
        ],
        "AllowedOAuthFlowsUserPoolClient": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeUserPoolClientà la section Référence des AWS CLI commandes.

`describe-user-pool-domain`

L'exemple de code suivant montre comment utiliser`describe-user-pool-domain`.

AWS CLI

Pour décrire un client de groupe d'utilisateurs

Cet exemple décrit un domaine de groupe d'utilisateurs nommé my-domain.

Commande :

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

Sortie :

```
{
    "DomainDescription": {
        "UserPoolId": "us-west-2_aaaaaaaaa",
        "AWSAccountId": "111111111111",
        "Domain": "my-domain",
        "S3Bucket": "aws-cognito-prod-pdx-assets",
        "CloudFrontDistribution": "aaaaaaaaaaaaa.cloudfront.net",
        "Version": "20190128175402",
        "Status": "ACTIVE",
        "CustomDomainConfig": {}
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeUserPoolDomainà la section Référence des AWS CLI commandes.

`describe-user-pool`

L'exemple de code suivant montre comment utiliser`describe-user-pool`.

AWS CLI

Pour décrire un groupe d'utilisateurs

L'exemple suivant décrit un groupe d'utilisateurs dont l'identifiant est US-West-2_Example.

```
aws cognito-idp describe-user-pool \
    --user-pool-id us-west-2_EXAMPLE
```

Sortie :

```
{
    "UserPool": {
        "Id": "us-west-2_EXAMPLE",
        "Name": "MyUserPool",
        "Policies": {
            "PasswordPolicy": {
                "MinimumLength": 8,
                "RequireUppercase": true,
                "RequireLowercase": true,
                "RequireNumbers": true,
                "RequireSymbols": true,
                "TemporaryPasswordValidityDays": 1
            }
        },
        "DeletionProtection": "ACTIVE",
        "LambdaConfig": {
            "PreSignUp": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreSignUpFunction",
            "CustomMessage": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomMessageFunction",
            "PostConfirmation": "arn:aws:lambda:us-
west-2:123456789012:function:MyPostConfirmationFunction",
```

```
            "PreAuthentication": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreAuthenticationFunction",
            "PostAuthentication": "arn:aws:lambda:us-
west-2:123456789012:function:MyPostAuthenticationFunction",
            "DefineAuthChallenge": "arn:aws:lambda:us-
west-2:123456789012:function:MyDefineAuthChallengeFunction",
            "CreateAuthChallenge": "arn:aws:lambda:us-
west-2:123456789012:function:MyCreateAuthChallengeFunction",
            "VerifyAuthChallengeResponse": "arn:aws:lambda:us-
west-2:123456789012:function:MyVerifyAuthChallengeFunction",
            "PreTokenGeneration": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreTokenGenerationFunction",
            "UserMigration": "arn:aws:lambda:us-
west-2:123456789012:function:MyMigrateUserFunction",
            "PreTokenGenerationConfig": {
                "LambdaVersion": "V2_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreTokenGenerationFunction"
            },
            "CustomSMSSender": {
                "LambdaVersion": "V1_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomSMSSenderFunction"
            },
            "CustomEmailSender": {
                "LambdaVersion": "V1_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomEmailSenderFunction"
            },
            "KMSKeyID": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
        },
        "LastModifiedDate": 1726784814.598,
        "CreationDate": 1602103465.273,
        "SchemaAttributes": [
            {
                "Name": "sub",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": false,
                "Required": true,
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "2048"
```

```
            }
        },
        {
            "Name": "name",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "given_name",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "family_name",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "middle_name",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
```

```
            }
        },
        {
            "Name": "nickname",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "preferred_username",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "profile",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "picture",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
```

```
            }
        },
        {
            "Name": "website",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "email",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": true,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "email_verified",
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false
        },
        {
            "Name": "gender",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "birthdate",
```

```
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "10",
                    "MaxLength": "10"
                }
            },
            {
                "Name": "zoneinfo",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "locale",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "phone_number",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "phone_number_verified",
```

```
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false
        },
        {
            "Name": "address",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "updated_at",
            "AttributeDataType": "Number",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "NumberAttributeConstraints": {
                "MinValue": "0"
            }
        },
        {
            "Name": "identities",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {}
        },
        {
            "Name": "custom:111",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "1",
                "MaxLength": "256"
            }
```

```
            },
            {
                "Name": "dev:custom:222",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": true,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "421"
                }
            },
            {
                "Name": "custom:accesstoken",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "custom:idtoken",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MaxLength": "2048"
                }
            }
        ],
        "AutoVerifiedAttributes": [
            "email"
        ],
        "SmsVerificationMessage": "Your verification code is {####}. ",
        "EmailVerificationMessage": "Your verification code is {####}. ",
        "EmailVerificationSubject": "Your verification code",
        "VerificationMessageTemplate": {
            "SmsMessage": "Your verification code is {####}. ",
            "EmailMessage": "Your verification code is {####}. ",
            "EmailSubject": "Your verification code",
```

```
            "EmailMessageByLink": "Please click the link below to verify your email
 address. <b>{##Verify Your Email##}</b>\n this is from us-west-2_ywDJHlIfU",
            "EmailSubjectByLink": "Your verification link",
            "DefaultEmailOption": "CONFIRM_WITH_LINK"
        },
        "SmsAuthenticationMessage": "Your verification code is {####}. ",
        "UserAttributeUpdateSettings": {
            "AttributesRequireVerificationBeforeUpdate": []
        },
        "MfaConfiguration": "OPTIONAL",
        "DeviceConfiguration": {
            "ChallengeRequiredOnNewDevice": true,
            "DeviceOnlyRememberedOnUserPrompt": false
        },
        "EstimatedNumberOfUsers": 166,
        "EmailConfiguration": {
            "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com",
            "EmailSendingAccount": "DEVELOPER"
        },
        "SmsConfiguration": {
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/userpool-
SMS-Role",
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsRegion": "us-west-2"
        },
        "UserPoolTags": {},
        "Domain": "myCustomDomain",
        "CustomDomain": "auth.example.com",
        "AdminCreateUserConfig": {
            "AllowAdminCreateUserOnly": false,
            "UnusedAccountValidityDays": 1,
            "InviteMessageTemplate": {
                "SMSMessage": "Your username is {username} and temporary password is
 {####}. ",
                "EmailMessage": "Your username is {username} and temporary password
 is {####}. ",
                "EmailSubject": "Your temporary password"
            }
        },
        "UserPoolAddOns": {
            "AdvancedSecurityMode": "ENFORCED",
            "AdvancedSecurityAdditionalFlows": {}
        },
```

```
        "Arn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE",
        "AccountRecoverySetting": {
            "RecoveryMechanisms": [
                {
                    "Priority": 1,
                    "Name": "verified_email"
                }
            ]
        }
    }
}
```

Pour plus d'informations sur les Groupes d'utilisateurs Amazon Cognito, veuillez consulter le Guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous DescribeUserPoolà la section Référence des AWS CLI commandes.

`forget-device`

L'exemple de code suivant montre comment utiliser`forget-device`.

AWS CLI

Pour oublier un appareil

Cet exemple oublie un appareil.

Commande :

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- Pour plus de détails sur l'API, reportez-vous ForgetDeviceà la section Référence des AWS CLI commandes.

`forgot-password`

L'exemple de code suivant montre comment utiliser`forgot-password`.

AWS CLI

Pour forcer le changement de mot de passe

L'`forgot-password`exemple suivant envoie un message à jane@example.com pour modifier son mot de passe.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --
username jane@example.com
```

Sortie :

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ForgotPasswordà la section Référence des AWS CLI commandes.

`get-csv-header`

L'exemple de code suivant montre comment utiliser`get-csv-header`.

AWS CLI

Pour créer un en-tête CSV

Cet exemple crée un en-tête CSV.

Pour plus d'informations sur l'importation d'utilisateurs, voir Importation d'utilisateurs dans des groupes d'utilisateurs à partir d'un fichier CSV.

Commande :

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaaa
```

Sortie :

```
{
  "UserPoolId": "us-west-2_aaaaaaaaa",
  "CSVHeader": [
      "name",
      "given_name",
      "family_name",
      "middle_name",
      "nickname",
      "preferred_username",
      "profile",
      "picture",
      "website",
      "email",
      "email_verified",
      "gender",
      "birthdate",
      "zoneinfo",
      "locale",
      "phone_number",
      "phone_number_verified",
      "address",
      "updated_at",
      "cognito:mfa_enabled",
      "cognito:username"
  ]
}
```

... Importation d'utilisateurs dans des groupes d'utilisateurs à partir d'un fichier CSV : https://docs.aws.amazon.com/cognito/ latest/developerguide/cognito - user-pools-using-import -tool.html

- Pour plus de détails sur l'API, reportez-vous GetCsvHeaderà la section Référence des AWS CLI commandes.

`get-device`

L'exemple de code suivant montre comment utiliser`get-device`.

AWS CLI

Pour obtenir un appareil

L'`get-device`exemple suivant affiche un appareil pour l'utilisateur actuellement connecté.

```
aws cognito-idp get-device \
    --access-token eyJra456defEXAMPLE \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "Device": {
        "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeviceAttributes": [
            {
                "Name": "device_status",
                "Value": "valid"
            },
            {
                "Name": "device_name",
                "Value": "MyDevice"
            },
            {
                "Name": "dev:device_arn",
                "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/diego.us-
west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            {
                "Name": "dev:device_owner",
                "Value": "diego.us-west-2_EXAMPLE"
            },
            {
                "Name": "last_ip_used",
                "Value": "192.0.2.1"
            },
            {
                "Name": "dev:device_remembered_status",
                "Value": "remembered"
            },
            {
                "Name": "dev:device_sdk",
                "Value": "aws-sdk"
            }
        ],
        "DeviceCreateDate": 1715100742.022,
```

```
        "DeviceLastModifiedDate": 1723233651.167,
        "DeviceLastAuthenticatedDate": 1715100742.0
    }
}
```

Pour plus d'informations, consultez la section Utilisation des appareils utilisateur de votre groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetDeviceà la section Référence des AWS CLI commandes.

`get-group`

L'exemple de code suivant montre comment utiliser`get-group`.

AWS CLI

Pour obtenir des informations sur un groupe

L'`get-group`exemple suivant répertorie les propriétés du groupe d'utilisateurs nomméMyGroup. Ce groupe possède une priorité et un rôle IAM lui est associé.

```
aws cognito-idp get-group \
    --user-pool-id us-west-2_EXAMPLE \
    --group-name MyGroup
```

Sortie :

```
{
    "Group": {
        "GroupName": "MyGroup",
        "UserPoolId": "us-west-2_EXAMPLE",
        "RoleArn": "arn:aws:iam::123456789012:role/example-cognito-role",
        "Precedence": 7,
        "LastModifiedDate": 1697211218.305,
        "CreationDate": 1611685503.954
    }
}
```

Pour plus d'informations, consultez la section Ajouter des groupes à un groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetGroupà la section Référence des AWS CLI commandes.

`get-identity-provider-by-identifier`

L'exemple de code suivant montre comment utiliser`get-identity-provider-by-identifier`.

AWS CLI

Pour obtenir la configuration d'un fournisseur d'identité à partir de l'identifiant IdP

L'`get-identity-provider-by-identifier`exemple suivant renvoie la configuration du fournisseur d'identité avec l'identifiant`mysso`.

```
aws cognito-idp get-identity-provider-by-identifier \
    --user-pool-id us-west-2_EXAMPLE \
    --idp-identifier mysso
```

Sortie :

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MYSAML",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate": "[Certificate contents]",
            "IDPSignout": "false",
            "MetadataURL": "https://auth.example.com/saml/metadata/",
            "SLORedirectBindingURI": "https://auth.example.com/saml/logout/",
            "SSORedirectBindingURI": "https://auth.example.com/saml/assertion/"
        },
        "AttributeMapping": {
            "email": "email"
        },
        "IdpIdentifiers": [
            "mysso",
            "mysamlsso"
        ],
        "LastModifiedDate": 1705616729.188,
        "CreationDate": 1643734622.919
    }
```

```
}
```

Pour plus d'informations, consultez la section Connexion à un IdP tiers dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetIdentityProviderByIdentifierà la section Référence des AWS CLI commandes.

`get-log-delivery-configuration`

L'exemple de code suivant montre comment utiliser`get-log-delivery-configuration`.

AWS CLI

Pour afficher la configuration de livraison du journal

L'`get-log-delivery-configuration`exemple suivant affiche les paramètres d'exportation du journal du groupe d'utilisateurs demandé.

```
aws cognito-idp get-log-delivery-configuration \
    --user-pool-id us-west-2_EXAMPLE
```

Sortie :

```
{
    "LogDeliveryConfiguration": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "LogConfigurations": [
            {
                "LogLevel": "INFO",
                "EventSource": "userAuthEvents",
                "FirehoseConfiguration": {
                    "StreamArn": "arn:aws:firehose:us-
west-2:123456789012:deliverystream/my-test-deliverystream"
                }
            },
            {
                "LogLevel": "ERROR",
                "EventSource": "userNotification",
                "CloudWatchLogsConfiguration": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:my-message-delivery-logs"
```

```
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Exportation des journaux des groupes d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetLogDeliveryConfigurationà la section Référence des AWS CLI commandes.

`get-signing-certificate`

L'exemple de code suivant montre comment utiliser`get-signing-certificate`.

AWS CLI

Pour afficher le certificat de signature SAML

L'`get-signing-certificate`exemple suivant affiche le certificat de signature SAML 2.0 pour le groupe d'utilisateurs de la demande.

```
aws cognito-idp get-signing-certificate \
    --user-pool-id us-west-2_EXAMPLE
```

Sortie :

```
{
    "Certificate": "[Certificate content]"
}
```

Pour plus d'informations, consultez la section Signature et chiffrement SAML dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetSigningCertificateà la section Référence des AWS CLI commandes.

`get-ui-customization`

L'exemple de code suivant montre comment utiliser`get-ui-customization`.

AWS CLI

Pour afficher les paramètres classiques de personnalisation de l'interface utilisateur hébergée pour un client d'application

L'`get-ui-customization`exemple suivant affiche les paramètres classiques de personnalisation de l'interface utilisateur hébergée pour un client d'application qui n'hérite pas des paramètres du groupe d'utilisateurs.

```
aws cognito-idp get-ui-customization \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789
```

Sortie :

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientId": "1example23456789",
        "ImageUrl": "https://example.cloudfront.net/us-
west-2_EXAMPLE/1example23456789/20250115191928/assets/images/image.jpg",
        "CSS": "\n.logo-customizable {\n  max-width: 80%;\n  max-height: 30%;
\n}\n\n.banner-customizable {\n  padding: 25px 0px 25px 0px;\n  background-color:
 lightgray;\n}\n\n.label-customizable {\n  font-weight: 400;\n}\n\n.textDescription-
customizable {\n  padding-top: 100px;\n  padding-bottom: 10px;\n  display: block;
\n  font-size: 12px;\n}\n\n.idpDescription-customizable {\n  padding-top: 10px;\n
  padding-bottom: 10px;\n  display: block;\n  font-size: 16px;\n}\n\n.legalText-
customizable {\n  color: #747474;\n  font-size: 11px;\n}\n\n.submitButton-
customizable {\n  font-size: 14px;\n  font-weight: bold;\n  margin: 20px 0px
 10px 0px;\n  height: 50px;\n  width: 100%;\n  color: #fff;\n  background-color:
 #337ab7;\n}\n\n.submitButton-customizable:hover {\n  color: #fff;\n  background-
color: #286090;\n}\n\n.errorMessage-customizable {\n  padding: 5px;\n  font-size:
 12px;\n  width: 100%;\n  background: #F5F5F5;\n  border: 2px solid #D64958;\n
  color: #D64958;\n}\n\n.inputField-customizable {\n  width: 100%;\n  height:
 34px;\n  color: #555;\n  background-color: #fff;\n  border: 1px solid #ccc;\n}\n
\n.inputField-customizable:focus {\n  border-color: #66afe9;\n  outline: 0;\n}\n
\n.idpButton-customizable {\n  height: 40px;\n  width: 100%;\n  width: 100%;\n
 text-align: center;\n  margin-bottom: 15px;\n  color: #fff;\n  background-color:
 #5bc0de;\n  border-color: #46b8da;\n}\n\n.idpButton-customizable:hover {\n
 color: #fff;\n  background-color: #31b0d5;\n}\n\n.socialButton-customizable {\n
  border-radius: 2px;\n  height: 60px;\n  margin-bottom: 15px;\n  padding: 1px;
\n  text-align: left;\n  width: 100%;\n}\n\n.redirect-customizable {\n  text-
align: center;\n}\n\n.passwordCheck-notValid-customizable {\n  color: #DF3312;
```

```
\n}\n\n.passwordCheck-valid-customizable {\n  color: #19BF00;\n}\n\n.background-
customizable {\n  background-color: #fff;\n}\n",
        "CSSVersion": "20250115191928"
    }
}
```

Pour plus d'informations, consultez la section Identification de l'interface utilisateur hébergée (classique) dans le guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous GetUiCustomizationà la section Référence des AWS CLI commandes.

`get-user-attribute-verification-code`

L'exemple de code suivant montre comment utiliser`get-user-attribute-verification-code`.

AWS CLI

Pour envoyer un code de vérification d'attribut à l'utilisateur actuel

L'`get-user-attribute-verification-code`exemple suivant envoie un code de vérification d'attribut à l'adresse e-mail de l'utilisateur actuellement connecté.

```
aws cognito-idp get-user-attribute-verification-code \
    --access-token eyJra456defEXAMPLE \
    --attribute-name email
```

Sortie :

```
{
    "CodeDeliveryDetails": {
        "Destination": "a***@e***",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

Pour plus d'informations, consultez Inscription et confirmation de comptes utilisateur dans le Guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous GetUserAttributeVerificationCodeà la section Référence des AWS CLI commandes.

`get-user-auth-factors`

L'exemple de code suivant montre comment utiliser`get-user-auth-factors`.

AWS CLI

Pour répertorier les facteurs d'authentification disponibles pour l'utilisateur actuel

L'`get-user-auth-factors`exemple suivant répertorie les facteurs d'authentification disponibles pour l'utilisateur actuellement connecté.

```
aws cognito-idp get-user-auth-factors \
    --access-token eyJra456defEXAMPLE
```

Sortie :

```
{
    "Username": "testuser",
    "ConfiguredUserAuthFactors": [
        "PASSWORD",
        "EMAIL_OTP",
        "SMS_OTP",
        "WEB_AUTHN"
    ]
}
```

Pour plus d'informations, consultez la section Authentification dans le guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous GetUserAuthFactorsà la section Référence des AWS CLI commandes.

`get-user-pool-mfa-config`

L'exemple de code suivant montre comment utiliser`get-user-pool-mfa-config`.

AWS CLI

Pour afficher l'authentification multifactorielle et WebAuthn les paramètres d'un groupe d'utilisateurs

L'`get-user-pool-mfa-config`exemple suivant montre le MFA et la WebAuthn configuration du groupe d'utilisateurs demandé.

```
aws cognito-idp get-user-pool-mfa-config \
    --user-pool-id us-west-2_EXAMPLE
```

Sortie :

```
{
    "SmsMfaConfiguration": {
        "SmsAuthenticationMessage": "Your OTP for MFA or sign-in: use {####}.",
        "SmsConfiguration": {
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/my-SMS-
Role",
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsRegion": "us-west-2"
        }
    },
    "SoftwareTokenMfaConfiguration": {
        "Enabled": true
    },
    "EmailMfaConfiguration": {
        "Message": "Your OTP for MFA or sign-in: use {####}",
        "Subject": "OTP test"
    },
    "MfaConfiguration": "OPTIONAL",
    "WebAuthnConfiguration": {
        "RelyingPartyId": "auth.example.com",
        "UserVerification": "preferred"
    }
}
```

Pour plus d'informations, consultez la section Ajouter une MFA dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetUserPoolMfaConfigà la section Référence des AWS CLI commandes.

`get-user`

L'exemple de code suivant montre comment utiliser`get-user`.

AWS CLI

Pour obtenir les détails de l'utilisateur actuel

L'`get-user`exemple suivant affiche le profil de l'utilisateur actuellement connecté.

```
aws cognito-idp get-user \
    --access-token eyJra456defEXAMPLE
```

Sortie :

```
{
    "Username": "johndoe",
    "UserAttributes": [
        {
            "Name": "sub",
            "Value": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
        {
            "Name": "identities",
            "Value": "[{\"userId\":\"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222\",
\"providerName\":\"SignInWithApple\",\"providerType\":\"SignInWithApple\",\"issuer
\":null,\"primary\":false,\"dateCreated\":1701125599632}]"
        },
        {
            "Name": "email_verified",
            "Value": "true"
        },
        {
            "Name": "custom:state",
            "Value": "Maine"
        },
        {
            "Name": "name",
            "Value": "John Doe"
        },
        {
            "Name": "phone_number_verified",
            "Value": "true"
        },
        {
            "Name": "phone_number",
            "Value": "+12065551212"
```

```
        },
        {
            "Name": "preferred_username",
            "Value": "jamesdoe"
        },
        {
            "Name": "locale",
            "Value": "EMEA"
        },
        {
            "Name": "email",
            "Value": "jamesdoe@example.com"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetUserà la section Référence des AWS CLI commandes.

`global-sign-out`

L'exemple de code suivant montre comment utiliser`global-sign-out`.

AWS CLI

Pour déconnecter l'utilisateur actuel

L'`global-sign-out`exemple suivant déconnecte l'utilisateur actuel.

```
aws cognito-idp global-sign-out \
    --access-token eyJra456defEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GlobalSignOutà la section Référence des AWS CLI commandes.

`initiate-auth`

L'exemple de code suivant montre comment utiliser`initiate-auth`.

AWS CLI

Pour connecter un utilisateur

L'`initiate-auth`exemple suivant permet de connecter un utilisateur avec le flux de base nom d'utilisateur/mot de passe, sans difficulté supplémentaire.

```
aws cognito-idp initiate-auth \
    --auth-flow USER_PASSWORD_AUTH \
    --client-id 1example23456789 \
    --analytics-metadata AnalyticsEndpointId=d70b2ba36a8c4dc5a04a0451aEXAMPLE \
    --auth-parameters USERNAME=testuser,PASSWORD=[Password] --user-context-
data EncodedData=mycontextdata --client-metadata MyTestKey=MyTestValue
```

Sortie :

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

Pour plus d'informations, consultez la section Authentification dans le guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous InitiateAuthà la section Référence des AWS CLI commandes.

`list-devices`

L'exemple de code suivant montre comment utiliser`list-devices`.

AWS CLI

Pour répertorier les appareils d'un utilisateur

L'`list-devices`exemple suivant répertorie les appareils enregistrés par l'utilisateur actuel.

```
aws cognito-idp list-devices \
    --access-token eyJra456defEXAMPLE
```

Sortie :

```
{
    "Devices": [
        {
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "device_name",
                    "Value": "Dart-device"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.1"
                }
            ],
            "DeviceCreateDate": 1715100742.022,
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceLastAuthenticatedDate": 1715100742.0,
            "DeviceLastModifiedDate": 1723233651.167
        },
        {
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
```

```
                    "Name": "last_ip_used",
                    "Value": "192.0.2.2"
                }
            ],
            "DeviceCreateDate": 1726856147.993,
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "DeviceLastAuthenticatedDate": 1726856147.0,
            "DeviceLastModifiedDate": 1726856147.993
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des appareils dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDevicesà la section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour répertorier les groupes d'un groupe d'utilisateurs

L'`list-groups`exemple suivant répertorie les deux premiers groupes du groupe d'utilisateurs demandé.

```
aws cognito-idp list-groups \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 2
```

Sortie :

```
{
    "Groups": [
        {
            "CreationDate": 1681760899.633,
            "Description": "My test group",
            "GroupName": "testgroup",
            "LastModifiedDate": 1681760899.633,
```

```
            "Precedence": 1,
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CreationDate": 1642632749.051,
            "Description": "Autogenerated group for users who sign in using
 Facebook",
            "GroupName": "us-west-2_EXAMPLE_Facebook",
            "LastModifiedDate": 1642632749.051,
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

Pour plus d'informations, consultez la section Ajouter des groupes à un groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListGroupsà la section Référence des AWS CLI commandes.

`list-identity-providers`

L'exemple de code suivant montre comment utiliser`list-identity-providers`.

AWS CLI

Pour répertorier les fournisseurs d'identité

L'`list-identity-providers`exemple suivant répertorie les deux premiers fournisseurs d'identité du groupe d'utilisateurs demandé.

```
aws cognito-idp list-identity-providers \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 2
```

Sortie :

```
{
    "Providers": [
        {
            "CreationDate": 1619477386.504,
            "LastModifiedDate": 1703798328.142,
```

```
            "ProviderName": "Azure",
            "ProviderType": "SAML"
        },
        {
            "CreationDate": 1642698776.175,
            "LastModifiedDate": 1642699086.453,
            "ProviderName": "LoginWithAmazon",
            "ProviderType": "LoginWithAmazon"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

Pour plus d'informations, consultez la section Connexion à un IdP tiers dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListIdentityProvidersà la section Référence des AWS CLI commandes.

`list-resource-servers`

L'exemple de code suivant montre comment utiliser`list-resource-servers`.

AWS CLI

Pour répertorier les serveurs de ressources

L'`list-resource-servers`exemple suivant répertorie les deux premiers serveurs de ressources du groupe d'utilisateurs demandé.

```
aws cognito-idp list-resource-servers \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 2
```

Sortie :

```
{
    "ResourceServers": [
        {
            "Identifier": "myapi.example.com",
            "Name": "Example API with custom access control scopes",
            "Scopes": [
                {
```

```
                    "ScopeDescription": "International customers",
                    "ScopeName": "international.read"
                },
                {
                    "ScopeDescription": "Domestic customers",
                    "ScopeName": "domestic.read"
                }
            ],
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "Identifier": "myapi2.example.com",
            "Name": "Another example API for access control",
            "Scopes": [
                {
                    "ScopeDescription": "B2B customers",
                    "ScopeName": "b2b.read"
                }
            ],
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

Pour plus d'informations, consultez la section Contrôle d'accès avec les serveurs de ressources dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListResourceServersà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises du groupe d'utilisateurs

L'`list-tags-for-resource`exemple suivant répertorie les balises attribuées au groupe d'utilisateurs avec l'ARN demandé.

```
aws cognito-idp list-tags-for-resource \
```

```
    --resource-arn arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE
```

Sortie :

```
{
    "Tags": {
        "administrator": "Jie",
        "tenant": "ExampleCorp"
    }
}
```

Pour plus d'informations, consultez la section Marquage des ressources Amazon Cognito dans le manuel du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-user-import-jobs`

L'exemple de code suivant montre comment utiliser`list-user-import-jobs`.

AWS CLI

Pour répertorier les tâches et les statuts d'importation des utilisateurs

L'`list-user-import-jobs`exemple suivant répertorie les trois premières tâches d'importation d'utilisateurs et leurs détails dans le groupe d'utilisateurs demandé.

```
aws cognito-idp list-user-import-jobs \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 3
```

Sortie :

```
{
    "PaginationToken": "us-west-2_EXAMPLE#import-example3#1667948397084",
    "UserImportJobs": [
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1735329786.142,
```

```
            "CompletionMessage": "The user import job has expired.",
            "CreationDate": 1735241621.022,
            "FailedUsers": 0,
            "ImportedUsers": 0,
            "JobId": "import-example1",
            "JobName": "Test-import-job-1",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "Status": "Expired",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1681509058.408,
            "CompletionMessage": "Too many users have failed or been skipped during
 the import.",
            "CreationDate": 1681509001.477,
            "FailedUsers": 1,
            "ImportedUsers": 0,
            "JobId": "import-example2",
            "JobName": "Test-import-job-2",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "StartDate": 1681509057.965,
            "Status": "Failed",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1.667864578676E9,
            "CompletionMessage": "Import Job Completed Successfully.",
            "CreationDate": 1.667864480281E9,
            "FailedUsers": 0,
            "ImportedUsers": 6,
```

```
            "JobId": "import-example3",
            "JobName": "Test-import-job-3",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "StartDate": 1.667864578167E9,
            "Status": "Succeeded",
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Importation d'utilisateurs depuis un fichier CSV dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListUserImportJobsà la section Référence des AWS CLI commandes.

`list-user-pool-clients`

L'exemple de code suivant montre comment utiliser`list-user-pool-clients`.

AWS CLI

Pour répertorier les clients de l'application

L'`list-user-pool-clients`exemple suivant répertorie les trois premiers clients d'applications du groupe d'utilisateurs demandé.

```
aws cognito-idp list-user-pool-clients \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 3
```

Sortie :

```
{
    "NextToken": "[Pagination token]",
    "UserPoolClients": [
        {
```

```
            "ClientId": "1example23456789",
            "ClientName": "app-client-1",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "ClientId": "2example34567890",
            "ClientName": "app-client-2",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "ClientId": "3example45678901",
            "ClientName": "app-client-3",
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Clients de l'application dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListUserPoolClientsà la section Référence des AWS CLI commandes.

`list-user-pools`

L'exemple de code suivant montre comment utiliser`list-user-pools`.

AWS CLI

Pour afficher les groupes d’utilisateurs

L'`list-user-pools`exemple suivant répertorie 3 des groupes d'utilisateurs disponibles dans le AWS compte des informations d'identification CLI actuelles.

```
aws cognito-idp list-user-pools \
    --max-results 3
```

Sortie :

```
{
    "NextToken": "[Pagination token]",
    "UserPools": [
        {
```

```
            "CreationDate": 1681502497.741,
            "Id": "us-west-2_EXAMPLE1",
            "LambdaConfig": {
                "CustomMessage": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreSignUp": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGeneration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGenerationConfig": {
                    "LambdaArn": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                    "LambdaVersion": "V1_0"
                }
            },
            "LastModifiedDate": 1681502497.741,
            "Name": "user pool 1"
        },
        {
            "CreationDate": 1686064178.717,
            "Id": "us-west-2_EXAMPLE2",
            "LambdaConfig": {
            },
            "LastModifiedDate": 1686064178.873,
            "Name": "user pool 2"
        },
        {
            "CreationDate": 1627681712.237,
            "Id": "us-west-2_EXAMPLE3",
            "LambdaConfig": {
                "UserMigration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction"
            },
            "LastModifiedDate": 1678486942.479,
            "Name": "user pool 3"
        }
    ]
}
```

Pour plus d'informations sur les Groupes d'utilisateurs Amazon Cognito, veuillez consulter le Guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous ListUserPoolsà la section Référence des AWS CLI commandes.

`list-users-in-group`

L'exemple de code suivant montre comment utiliser`list-users-in-group`.

AWS CLI

Pour répertorier les utilisateurs d'un groupe

Cet exemple répertorie les utilisateurs dans un groupe MyGroup.

Commande :

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyGroup
```

Sortie :

```
{
  "Users": [
      {
          "Username": "acf10624-80bb-401a-ac61-607bee2110ec",
          "Attributes": [
              {
                  "Name": "sub",
                  "Value": "acf10624-80bb-401a-ac61-607bee2110ec"
              },
              {
                  "Name": "custom:CustomAttr1",
                  "Value": "New Value!"
              },
              {
                  "Name": "email",
                  "Value": "jane@example.com"
              }
          ],
          "UserCreateDate": 1548102770.284,
          "UserLastModifiedDate": 1548103204.893,
          "Enabled": true,
          "UserStatus": "CONFIRMED"
      },
      {
          "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
          "Attributes": [
              {
```

```
                "Name": "sub",
                "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
            },
            {
                "Name": "email_verified",
                "Value": "true"
            },
            {
                "Name": "email",
                "Value": "diego@example.com"
            }
        ],
        "UserCreateDate": 1548089817.683,
        "UserLastModifiedDate": 1548089817.683,
        "Enabled": true,
        "UserStatus": "FORCE_CHANGE_PASSWORD"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListUsersInGroupà la section Référence des AWS CLI commandes.

`list-users`

L'exemple de code suivant montre comment utiliser`list-users`.

AWS CLI

Exemple 1 : pour répertorier les utilisateurs avec un filtre côté serveur

L'`list-users`exemple suivant répertorie 3 utilisateurs du groupe d'utilisateurs demandé dont les adresses e-mail commencent par`testuser`.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --filter email^=\"testuser\" \
    --max-items 3
```

Sortie :

```
{
```

```
    "PaginationToken": "efgh5678EXAMPLE",
    "Users": [
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "eaad0219-2117-439f-8d46-4db20e59268f"
                },
                {
                    "Name": "email",
                    "Value": "testuser@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1682955829.578,
            "UserLastModifiedDate": 1689030181.63,
            "UserStatus": "CONFIRMED",
            "Username": "testuser"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "3b994cfd-0b07-4581-be46-3c82f9a70c90"
                },
                {
                    "Name": "email",
                    "Value": "testuser2@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427979.201,
            "UserLastModifiedDate": 1684427979.201,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser2"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "5929e0d1-4c34-42d1-9b79-a5ecacfe66f7"
                },
                {
                    "Name": "email",
```

```
                    "Value": "testuser3@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427823.641,
            "UserLastModifiedDate": 1684427823.641,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser3@example.com"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion et recherche d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

Exemple 2 : pour répertorier les utilisateurs avec un filtre côté client

L'`list-users`exemple suivant répertorie les attributs de trois utilisateurs dont un attribut, en l'occurrence leur adresse e-mail, contient le domaine de messagerie "@example .com ». Si d'autres attributs contenaient cette chaîne, ils seraient également affichés. Le second utilisateur n'a aucun attribut correspondant à la requête et est exclu de la sortie affichée, mais pas de la réponse du serveur.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 3
    --query Users\[\*\].Attributes\[\?Value\.contains\(\@\,\'@example.com\'\)\]
```

Sortie :

```
[
    [
        {
            "Name": "email",
            "Value": "admin@example.com"
        }
    ],
    [],
    [
        {
            "Name": "email",
            "Value": "operator@example.com"
```

```
        }
    ]
]
```

Pour plus d'informations, consultez la section Gestion et recherche d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`list-web-authn-credentials`

L'exemple de code suivant montre comment utiliser`list-web-authn-credentials`.

AWS CLI

Pour répertorier les informations d'identification par clé d'accès

L'`list-web-authn-credentials`exemple suivant répertorie les clés d'accès ou les informations WebAuthn d'identification de l'utilisateur actuel. Ils ont un appareil enregistré.

```
aws cognito-idp list-web-authn-credentials \
    --access-token eyJra456defEXAMPLE
```

Sortie :

```
{
    "Credentials": [
        {
            "AuthenticatorAttachment": "cross-platform",
            "CreatedAt": 1736293876.115,
            "CredentialId": "8LApgk4-lNUFHbhm2w6Und7-
uxcc8coJGsPxiogvHoItc64xWQc3r4CEXAMPLE",
            "FriendlyCredentialName": "Roaming passkey",
            "RelyingPartyId": "auth.example.com"
        }
    ]
}
```

Pour plus d'informations, consultez la section Passkey login dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListWebAuthnCredentialsà la section Référence des AWS CLI commandes.

`resend-confirmation-code`

L'exemple de code suivant montre comment utiliser`resend-confirmation-code`.

AWS CLI

Pour renvoyer un code de confirmation

L'exemple `resend-confirmation-code` suivant envoie un code de confirmation à l'utilisateur `jane`.

```
aws cognito-idp resend-confirmation-code \
    --client-id 12a3b456c7de890f11g123hijk \
    --username jane
```

Sortie :

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

Pour plus d'informations, consultez Inscription et confirmation de comptes utilisateur dans le Guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous ResendConfirmationCodeà la section Référence des AWS CLI commandes.

`respond-to-auth-challenge`

L'exemple de code suivant montre comment utiliser`respond-to-auth-challenge`.

AWS CLI

Exemple 1 : pour répondre à un défi NEW_PASSWORD_REQUIRED

L'`respond-to-auth-challenge`exemple suivant répond à un défi NEW_PASSWORD_REQUIRED renvoyé par initiate-auth. Il définit un mot de passe pour l'utilisateur`jane@example.com`.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --challenge-name NEW_PASSWORD_REQUIRED \
    --challenge-responses USERNAME=jane@example.com,NEW_PASSWORD=[Password] \
    --session AYABeEv5HklEXAMPLE
```

Sortie :

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "ACCESS_TOKEN",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "REFRESH_TOKEN",
        "IdToken": "ID_TOKEN",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-wt2ha1Zd"
        }
    }
}
```

Pour plus d'informations, consultez la section Authentification dans le guide du développeur Amazon Cognito.

Exemple 2 : pour répondre à un défi SELECT_MFA_TYPE

L'`respond-to-auth-challenge`exemple suivant choisit TOTP MFA comme option MFA pour l'utilisateur actuel. L'utilisateur a été invité à sélectionner un type de MFA et sera ensuite invité à saisir son code MFA.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789
    --session AYABeEv5HklEXAMPLE
    --challenge-name SELECT_MFA_TYPE
    --challenge-responses USERNAME=testuser,ANSWER=SOFTWARE_TOKEN_MFA
```

Sortie :

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeEv5HklEXAMPLE",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "transparent"
    }
}
```

Pour plus d'informations, consultez la section Ajouter une MFA dans le manuel Amazon Cognito Developer Guide.

Exemple 3 : pour répondre à un défi SOFTWARE_TOKEN_MFA

L'`respond-to-auth-challenge`exemple suivant fournit un code MFA TOTP et termine la connexion.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --session AYABeEv5HklEXAMPLE \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=testuser,SOFTWARE_TOKEN_MFA_CODE=123456
```

Sortie :

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

Pour plus d'informations, consultez la section Ajouter une MFA dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RespondToAuthChallengeà la section Référence des AWS CLI commandes.

`revoke-token`

L'exemple de code suivant montre comment utiliser`revoke-token`.

AWS CLI

Pour révoquer un jeton d'actualisation

Ce qui suit `revoke-token` révoque le jeton d'actualisation demandé et les jetons d'accès associés.

```
aws cognito-idp revoke-token \
    --token eyJjd123abcEXAMPLE \
    --client-id 1example23456789
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Révoquer des jetons dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RevokeTokenà la section Référence des AWS CLI commandes.

`set-log-delivery-configuration`

L'exemple de code suivant montre comment utiliser`set-log-delivery-configuration`.

AWS CLI

Pour configurer l'exportation des journaux à partir d'un groupe d'utilisateurs

L'`set-log-delivery-configuration`exemple suivant configure le groupe d'utilisateurs demandé en enregistrant les erreurs de notification utilisateur dans un groupe de journaux et en enregistrant les informations d'authentification utilisateur dans un compartiment S3.

```
aws cognito-idp set-log-delivery-configuration \
    --user-pool-id us-west-2_EXAMPLE \
    --log-
configurations LogLevel=ERROR,EventSource=userNotification,CloudWatchLogsConfiguration={LogG
```

```
west-2:123456789012:log-group:cognito-
exported} LogLevel=INFO,EventSource=userAuthEvents,S3Configuration={BucketArn=arn:aws:s3:::a
s3-demo-bucket1}
```

Sortie :

```
{
   "LogDeliveryConfiguration": {
        "LogConfigurations": [
            {
                "CloudWatchLogsConfiguration": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:cognito-exported"
                },
                "EventSource": "userNotification",
                "LogLevel": "ERROR"
            },
            {
                "EventSource": "userAuthEvents",
                "LogLevel": "INFO",
                "S3Configuration": {
                    "BucketArn": "arn:aws:s3:::amzn-s3-demo-bucket1"
                }
            }
        ],
        "UserPoolId": "us-west-2_EXAMPLE"
   }
}
```

Pour plus d'informations, consultez la section Exportation des journaux des groupes d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetLogDeliveryConfigurationà la section Référence des AWS CLI commandes.

`set-risk-configuration`

L'exemple de code suivant montre comment utiliser`set-risk-configuration`.

AWS CLI

Pour définir la configuration des risques de protection contre les menaces

L'`set-risk-configuration`exemple suivant configure les messages et les actions de protection contre les menaces, les informations d'identification compromises et les exceptions d'adresse IP dans le client d'application demandé. En raison de la complexité de l' NotifyConfiguration objet, la saisie JSON est une bonne pratique pour cette commande.

```
aws cognito-idp set-risk-configuration \
    --cli-input-json file://set-risk-configuration.json
```

Contenu de `set-risk-configuration.json` :

```
{
    "AccountTakeoverRiskConfiguration": {
        "Actions": {
            "HighAction": {
                "EventAction": "MFA_REQUIRED",
                "Notify": true
            },
            "LowAction": {
                "EventAction": "NO_ACTION",
                "Notify": true
            },
            "MediumAction": {
                "EventAction": "MFA_IF_CONFIGURED",
                "Notify": true
            }
        },
        "NotifyConfiguration": {
            "BlockEmail": {
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "Subject": "Blocked sign-in attempt",
                "TextBody": "We blocked an unrecognized sign-in to your account with
 this information:\nTime: {login-time}\nDevice: {device-name}\nLocation: {city},
 {country}\nIf this sign-in was not by you, you should change your password and
 notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you, you
 can follow {one-click-link-valid} to let us know"
```

```
            },
            "From": "admin@example.com",
            "MfaEmail": {
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
 context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
 you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "Subject": "New sign-in attempt",
                "TextBody": "We required you to use multi-factor authentication
 for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-
name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should
 change your password and notify us by clicking on {one-click-link-invalid}\nIf this
 sign-in was by you, you can follow {one-click-link-valid} to let us know"
            },
            "NoActionEmail": {
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "Subject": "New sign-in attempt",
                "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
            },
            "ReplyTo": "admin@example.com",
            "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
        }
    },
    "ClientId": "1example23456789",
    "CompromisedCredentialsRiskConfiguration": {
        "Actions": {
            "EventAction": "BLOCK"
        },
```

```
        "EventFilter": [
            "PASSWORD_CHANGE",
            "SIGN_UP",
            "SIGN_IN"
        ]
    },
    "RiskExceptionConfiguration": {
        "BlockedIPRangeList": [
            "192.0.2.1/32",
            "192.0.2.2/32"
        ],
        "SkippedIPRangeList": [
            "203.0.113.1/32",
            "203.0.113.2/32"
        ]
    },
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

Sortie :

```
{
    "RiskConfiguration": {
        "AccountTakeoverRiskConfiguration": {
            "Actions": {
                "HighAction": {
                    "EventAction": "MFA_REQUIRED",
                    "Notify": true
                },
                "LowAction": {
                    "EventAction": "NO_ACTION",
                    "Notify": true
                },
                "MediumAction": {
                    "EventAction": "MFA_IF_CONFIGURED",
                    "Notify": true
                }
            },
            "NotifyConfiguration": {
                "BlockEmail": {
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
```

```
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "Subject": "Blocked sign-in attempt",
                    "TextBody": "We blocked an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
                },
                "From": "admin@example.com",
                "MfaEmail": {
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 required you to use multi-factor authentication for the following sign-in attempt:
\n<ul>\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location:
 {city}, {country}</li>\n</ul>\nIf this sign-in was not by you, you should change
 your password and notify us by clicking on <a href={one-click-link-invalid}>this
 link</a>\nIf this sign-in was by you, you can follow <a href={one-click-link-
valid}>this link</a> to let us know</pre>\n</body>\n</html>",
                    "Subject": "New sign-in attempt",
                    "TextBody": "We required you to use multi-factor authentication
 for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-
name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should
 change your password and notify us by clicking on {one-click-link-invalid}\nIf this
 sign-in was by you, you can follow {one-click-link-valid} to let us know"
                },
                "NoActionEmail": {
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "Subject": "New sign-in attempt",
                    "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
```

```
                },
                "ReplyTo": "admin@example.com",
                "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
            }
        },
        "ClientId": "1example23456789",
        "CompromisedCredentialsRiskConfiguration": {
            "Actions": {
                "EventAction": "BLOCK"
            },
            "EventFilter": [
                "PASSWORD_CHANGE",
                "SIGN_UP",
                "SIGN_IN"
            ]
        },
        "RiskExceptionConfiguration": {
            "BlockedIPRangeList": [
                "192.0.2.1/32",
                "192.0.2.2/32"
            ],
            "SkippedIPRangeList": [
                "203.0.113.1/32",
                "203.0.113.2/32"
            ]
        },
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Protection contre les menaces dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetRiskConfigurationà la section Référence des AWS CLI commandes.

`set-ui-customization`

L'exemple de code suivant montre comment utiliser`set-ui-customization`.

AWS CLI

Exemple 1 : pour personnaliser l'interface utilisateur hébergée classique pour un client d'application

L'`set-ui-customization`exemple suivant configure le client d'application demandé avec du CSS personnalisé et avec le logo Amazon Cognito comme logo de l'application.

```
aws cognito-idp set-ui-customization \
    --user-pool-id us-west-2_ywDJHlIfU \
    --client-id 14pq32c5q2uq2q7keorloqvb23 \
    --css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-customizable
 {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-
size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n\tpadding-
bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-customizable
 {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable {\n\tfont-
size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;\n\theight:
 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;\n\ttext-align:
 center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
 #286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
 #D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
 #555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n\tborder-radius:
 0px;\n}\n.inputField-customizable:focus {\n\tborder-color: #66afe9;\n\toutline:
 0;\n}\n.idpButton-customizable {\n\theight: 40px;\n\twidth: 100%;\n\twidth: 100%;
\n\ttext-align: center;\n\tmargin-bottom: 15px;\n\tcolor: #fff;\n\tbackground-
color: #5bc0de;\n\tborder-color: #46b8da;\n}\n.idpButton-customizable:hover {\n
\tcolor: #fff;\n\tbackground-color: #31b0d5;\n}\n.socialButton-customizable {\n
\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-bottom: 15px;\n\tpadding: 1px;
\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #fff;\n}\n" \
    --image-
file iVBORw0KGgoAAAANSUhEUgAAAFAAAABQCAMAAAC5zwKfAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAA
Cmsfvm6f3y9P////fM0uqAj
+yNmu6ZpvnZ3eNabuFNYuZneehzhPKzvPTAxwAAAOiMMlkAAAASdFJOU//////////////////////
AOK/vxIAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAKDSURBVFhH7ZfpkoMgEISDHKuEw/
d/2u2BQWMiBrG29o+fVsKatdPMAeZxc3Nz8w+ISekzmB++sYIw/I/
tjHzrPpO2Tx62EbR2PNxFac+jVuKxRaV50IzXkUe76NOCoUuwlvnQKei02gNF0ykotOLRBq/
nboeWRxAISx2EbsHFoRhK6Igk2JJlwScfQjgt06dOaWWiTbEDAe/
```

```
iq8N9kqCw2uCbHkHlYkaXEF8EYeL9RDqT4FhC6XMIIEifdcUwCc4leNyhabadWU6OlKYJE1Oac3NSPhB5rlaXlSgmr/1
F0L6Q5pZiSG0SfZTSTCOUhxOCH1AdIoCpTTIjtd
+VpEjUDDytQH/0Fpc661Aisas/4qmyUItD557pSCOSQQzlx27J
+meyDGc5zZgfhWuXE1lGgmVOMwmWdeGdzhjqZV14x5vSj7vsC5JDz/Cl0Vhp56n2NQt1wQIpury1EPbwyaYm
+IhmAQKoajkH51wg4cMZ1wQ3QG9efKWWOaDhYWnU6jXjCMdRmm21PArI
+Pb5DYoH93hq0ZCPlxeGJho/DI15C6sQc/L2sTC47UFBKZGHT6k+zlXg7WebA0Nr0HTcLMfk/
Y4Rc65D3iG6WDd7YLSlVqk87bVhUwhnClrx11RsVQwlAA818Mn
+QEs71BhSFU6orsUfKhHp72XMGYXi4q9c64RXRvzkWurRfG2vI2be/VaNcNgpX0Evb/
vio7nPMmj5qujkpQgSaPd1UcVqciHFDNZpOcGlcOPyi+AamCbIL9fitxAGeFN2Dl
+3vZubm5u/4fH4Bd14HhIPdwZPAAAAAElFTkSuQmCC
```

Sortie :

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_ywDJHlIfU",
        "ClientId": "14pq32c5q2uq2q7keorloqvb23",
        "ImageUrl": "https://
cf.thewrong.club/14pq32c5q2uq2q7keorloqvb23/20250117005911/assets/images/image.jpg",
        "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
 {\n\tfont-size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;
\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n\ttext-align: center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;
\n\tbackground-color: #286090;\n}\n.errorMessage-customizable {\n\tpadding:
 5px;\n\tfont-size: 14px;\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder:
 2px solid #D64958;\n\tcolor: #D64958;\n}\n.inputField-customizable {\n\twidth:
 100%;\n\theight: 34px;\n\tcolor: #555;\n\tbackground-color: #fff;\n\tborder:
 1px solid #ccc;\n\tborder-radius: 0px;\n}\n.inputField-customizable:focus {\n
\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-customizable {\n\theight:
 40px;\n\twidth: 100%;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-bottom:
 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color: #46b8da;
\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color: #31b0d5;
\n}\n.socialButton-customizable {\n\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-
bottom: 15px;\n\tpadding: 1px;\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-
customizable {\n\ttext-align: center;\n}\n.passwordCheck-notValid-customizable
 {\n\tcolor: #DF3312;\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;
\n}\n.background-customizable {\n\tbackground-color: #fff;\n}\n",
```

```
        "CSSVersion": "20250117005911"
    }
}
```

Exemple 2 : pour définir la personnalisation de l'interface utilisateur par défaut pour tous les clients de l'application

L'`set-ui-customization`exemple suivant configure le groupe d'utilisateurs demandé pour tous les clients d'applications qui n'ont pas de configuration spécifique au client. La commande applique un code CSS personnalisé et utilise le logo Amazon Cognito comme logo de l'application.

```
aws cognito-idp set-ui-customization \
--user-pool-id us-west-2_ywDJHlIfU \
--client-id ALL \
--css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-customizable
 {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-
size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n\tpadding-
bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-customizable
 {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable {\n\tfont-
size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;\n\theight:
 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;\n\ttext-align:
 center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
 #286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
 #D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
 #555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n\tborder-radius:
 0px;\n}\n.inputField-customizable:focus {\n\tborder-color: #66afe9;\n\toutline:
 0;\n}\n.idpButton-customizable {\n\theight: 40px;\n\twidth: 100%;\n\twidth: 100%;
\n\ttext-align: center;\n\tmargin-bottom: 15px;\n\tcolor: #fff;\n\tbackground-
color: #5bc0de;\n\tborder-color: #46b8da;\n}\n.idpButton-customizable:hover {\n
\tcolor: #fff;\n\tbackground-color: #31b0d5;\n}\n.socialButton-customizable {\n
\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-bottom: 15px;\n\tpadding: 1px;
\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #fff;\n}\n" \
--image-
file iVBORw0KGgoAAAANSUhEUgAAAFAAAABQCAMAAAC5zwKfAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAA
Cmsfvm6f3y9P////fM0uqAj
+yNmu6ZpvnZ3eNabuFNYuZneehzhPKzvPTAxwAAAOiMMlkAAAASdFJOU//////////////////////
```

```
AOK/vxIAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAKDSURBVFhH7ZfpkoMgEISDHKuEw/
d/2u2BQWMiBrG29o+fVsKatdPMAeZxc3Nz8w+ISekzmB++sYIw/I/
tjHzrPpO2Tx62EbR2PNxFac+jVuKxRaV50IzXkUe76NOCoUuwlvnQKei02gNF0ykotOLRBq/
nboeWRxAISx2EbsHFoRhK6Igk2JJlwScfQjgt06dOaWWiTbEDAe/
iq8N9kqCw2uCbHkHlYkaXEF8EYeL9RDqT4FhC6XMIIEifdcUwCc4leNyhabadWU6OlKYJE1Oac3NSPhB5rlaXlSgmr/1
F0L6Q5pZiSG0SfZTSTCOUhxOCH1AdIoCpTTIjtd
+VpEjUDDytQH/0Fpc661Aisas/4qmyUItD557pSCOSQQzlx27J
+meyDGc5zZgfhWuXE1lGgmVOMwmWdeGdzhjqZV14x5vSj7vsC5JDz/Cl0Vhp56n2NQt1wQIpury1EPbwyaYm
+IhmAQKoajkH51wg4cMZ1wQ3QG9efKWWOaDhYWnU6jXjCMdRmm21PArI
+Pb5DYoH93hq0ZCPlxeGJho/DI15C6sQc/L2sTC47UFBKZGHT6k+zlXg7WebA0Nr0HTcLMfk/
Y4Rc65D3iG6WDd7YLSlVqk87bVhUwhnClrx11RsVQwlAA818Mn
+QEs71BhSFU6orsUfKhHp72XMGYXi4q9c64RXRvzkWurRfG2vI2be/VaNcNgpX0Evb/
vio7nPMmj5qujkpQgSaPd1UcVqciHFDNZpOcGlcOPyi+AamCbIL9fitxAGeFN2Dl
+3vZubm5u/4fH4Bd14HhIPdwZPAAAAAElFTkSuQmCC
```

Sortie :

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_ywDJHlIfU",
        "ClientId": "14pq32c5q2uq2q7keorloqvb23",
        "ImageUrl": "https://
cf.thewrong.club/14pq32c5q2uq2q7keorloqvb23/20250117005911/assets/images/image.jpg",
        "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
 {\n\tfont-size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;
\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n\ttext-align: center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;
\n\tbackground-color: #286090;\n}\n.errorMessage-customizable {\n\tpadding:
 5px;\n\tfont-size: 14px;\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder:
 2px solid #D64958;\n\tcolor: #D64958;\n}\n.inputField-customizable {\n\twidth:
 100%;\n\theight: 34px;\n\tcolor: #555;\n\tbackground-color: #fff;\n\tborder:
 1px solid #ccc;\n\tborder-radius: 0px;\n}\n.inputField-customizable:focus {\n
\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-customizable {\n\theight:
 40px;\n\twidth: 100%;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-bottom:
 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color: #46b8da;
\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color: #31b0d5;
\n}\n.socialButton-customizable {\n\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-
```

```
bottom: 15px;\n\tpadding: 1px;\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-
customizable {\n\ttext-align: center;\n}\n.passwordCheck-notValid-customizable
 {\n\tcolor: #DF3312;\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;
\n}\n.background-customizable {\n\tbackground-color: #fff;\n}\n",
        "CSSVersion": "20250117005911"
    }
}
```

Pour plus d'informations, consultez la section Identification de l'interface utilisateur hébergée (classique) dans le guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous SetUiCustomizationà la section Référence des AWS CLI commandes.

`set-user-mfa-preference`

L'exemple de code suivant montre comment utiliser`set-user-mfa-preference`.

AWS CLI

Pour définir les préférences MFA d'un utilisateur

L'`set-user-mfa-preference`exemple suivant configure l'utilisateur actuel pour utiliser le TOTP MFA et désactive tous les autres facteurs MFA.

```
aws cognito-idp set-user-mfa-preference \
    --access-token eyJra456defEXAMPLE \
    --software-token-mfa-settings Enabled=true,PreferredMfa=true \
    --sms-mfa-settings Enabled=false,PreferredMfa=false \
    --email-mfa-settings Enabled=false,PreferredMfa=false
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter une MFA dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetUserMfaPreferenceà la section Référence des AWS CLI commandes.

`set-user-pool-mfa-config`

L'exemple de code suivant montre comment utiliser`set-user-pool-mfa-config`.

AWS CLI

Pour configurer le MFA du pool d'utilisateurs et WebAuthn

L'`set-user-pool-mfa-config`exemple suivant configure le groupe d'utilisateurs demandé avec l'authentification MFA optionnelle avec toutes les méthodes MFA disponibles, et définit la configuration. WebAuthn

```
aws cognito-idp set-user-pool-mfa-config \
    --user-pool-id us-west-2_EXAMPLE \
    --sms-mfa-configuration "SmsAuthenticationMessage=\"Your OTP for MFA or sign-
in: use {####}.\",SmsConfiguration={SnsCallerArn=arn:aws:iam::123456789012:role/
service-role/test-SMS-Role,ExternalId=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111,SnsRegion=us-west-2}" \
    --software-token-mfa-configuration Enabled=true \
    --email-mfa-configuration "Message=\"Your OTP for MFA or sign-in: use
 {####}\",Subject=\"OTP test\"" \
    --mfa-configuration OPTIONAL \
    --web-authn-
configuration RelyingPartyId=auth.example.com,UserVerification=preferred
```

Sortie :

```
{
    "EmailMfaConfiguration": {
        "Message": "Your OTP for MFA or sign-in: use {####}",
        "Subject": "OTP test"
    },
    "MfaConfiguration": "OPTIONAL",
    "SmsMfaConfiguration": {
        "SmsAuthenticationMessage": "Your OTP for MFA or sign-in: use {####}.",
        "SmsConfiguration": {
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/test-SMS-
Role",
            "SnsRegion": "us-west-2"
        }
    },
    "SoftwareTokenMfaConfiguration": {
        "Enabled": true
    },
    "WebAuthnConfiguration": {
        "RelyingPartyId": "auth.example.com",
```

```
        "UserVerification": "preferred"
    }
}
```

Pour plus d'informations, consultez la section Ajout d'une authentification MFA et d'un mot de passe dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetUserPoolMfaConfigà la section Référence des AWS CLI commandes.

`set-user-settings`

L'exemple de code suivant montre comment utiliser`set-user-settings`.

AWS CLI

Pour définir les paramètres utilisateur

Cet exemple définit la préférence de livraison MFA sur EMAIL.

Commande :

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-
options DeliveryMedium=EMAIL
```

- Pour plus de détails sur l'API, reportez-vous SetUserSettingsà la section Référence des AWS CLI commandes.

`sign-up`

L'exemple de code suivant montre comment utiliser`sign-up`.

AWS CLI

Pour inscrire un utilisateur

Cet exemple inscrit jane@example.com.

Commande :

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username jane@example.com --password PASSWORD --user-attributes
 Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Sortie :

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- Pour plus de détails sur l'API, reportez-vous SignUpà la section Référence des AWS CLI commandes.

`start-user-import-job`

L'exemple de code suivant montre comment utiliser`start-user-import-job`.

AWS CLI

Pour démarrer une tâche d'importation

L'`start-user-import-job`exemple suivant démarre la tâche d'importation demandée dans le groupe d'utilisateurs demandé.

```
aws cognito-idp start-user-import-job \
    --user-pool-id us-west-2_EXAMPLE \
    --job-id import-mAgUtd8PMm
```

Sortie :

```
{
    "UserImportJob": {
        "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/example-cloudwatch-
logs-role",
        "CreationDate": 1736442975.904,
        "FailedUsers": 0,
        "ImportedUsers": 0,
        "JobId": "import-mAgUtd8PMm",
        "JobName": "Customer import",
```

```
        "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
        "SkippedUsers": 0,
        "StartDate": 1736443020.081,
        "Status": "Pending",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Importation d'utilisateurs dans un groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartUserImportJobà la section Référence des AWS CLI commandes.

`start-web-authn-registration`

L'exemple de code suivant montre comment utiliser`start-web-authn-registration`.

AWS CLI

Pour obtenir les informations d'enregistrement par clé d'accès pour un utilisateur connecté

L'`start-web-authn-registration`exemple suivant génère des options WebAuthn d'enregistrement pour l'utilisateur actuel.

```
aws cognito-idp start-web-authn-registration \
    --access-token eyJra456defEXAMPLE
```

Sortie :

```
{
    "CredentialCreationOptions": {
        "authenticatorSelection": {
            "requireResidentKey": true,
            "residentKey": "required",
            "userVerification": "preferred"
        },
        "challenge": "wxvbDicyqQqvF2EXAMPLE",
```

```
        "excludeCredentials": [
            {
                "id": "8LApgk4-lNUFHbhm2w6Und7-
uxcc8coJGsPxiogvHoItc64xWQc3r4CEXAMPLE",
                "type": "public-key"
            }
        ],
        "pubKeyCredParams": [
            {
                "alg": -7,
                "type": "public-key"
            },
            {
                "alg": -257,
                "type": "public-key"
            }
        ],
        "rp": {
            "id": "auth.example.com",
            "name": "auth.example.com"
        },
        "timeout": 60000,
        "user": {
            "displayName": "testuser",
            "id": "ZWFhZDAyMTktMjExNy00MzlmLThkNDYtNGRiMjBlNEXAMPLE",
            "name": "testuser"
        }
    }
}
```

Pour plus d'informations, consultez la section Passkey login dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartWebAuthnRegistrationà la section Référence des AWS CLI commandes.

`stop-user-import-job`

L'exemple de code suivant montre comment utiliser`stop-user-import-job`.

AWS CLI

Pour arrêter une tâche d'importation

L'`stop-user-import-job`exemple suivant arrête la tâche d'importation d'utilisateurs en cours d'exécution demandée dans le groupe d'utilisateurs demandé.

```
aws cognito-idp stop-user-import-job \
    --user-pool-id us-west-2_EXAMPLE \
    --job-id import-mAgUtd8PMm
```

Sortie :

```
{
    "UserImportJob": {
        "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/example-cloudwatch-
logs-role",
        "CompletionDate": 1736443496.379,
        "CompletionMessage": "The Import Job was stopped by the developer.",
        "CreationDate": 1736443471.781,
        "FailedUsers": 0,
        "ImportedUsers": 0,
        "JobId": "import-mAgUtd8PMm",
        "JobName": "Customer import",
        "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
        "SkippedUsers": 0,
        "StartDate": 1736443494.154,
        "Status": "Stopped",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Importation d'utilisateurs dans un groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopUserImportJobà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un groupe d'utilisateurs

L'`tag-resource`exemple suivant s'applique `administrator` et `department` attribue des balises au groupe d'utilisateurs demandé.

```
aws cognito-idp tag-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_EXAMPLE \
    --tags administrator=Jie,tenant=ExampleCorp
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon Cognito dans le manuel du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'un groupe d'utilisateurs

L'`untag-resource`exemple suivant supprime `administrator` et `department` étiquette du groupe d'utilisateurs demandé.

```
aws cognito-idp untag-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:767671399759:userpool/us-west-2_l5cxwdm2K \
    --tag-keys administrator tenant
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon Cognito dans le manuel du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-auth-event-feedback`

L'exemple de code suivant montre comment utiliser`update-auth-event-feedback`.

AWS CLI

Pour mettre à jour les commentaires relatifs aux événements d'authentification

Cet exemple met à jour les commentaires relatifs aux événements d'autorisation. Cela indique que l'événement est « valide ».

Commande :

```
aws cognito-idp update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --event-id EVENT_ID --feedback-token FEEDBACK_TOKEN --
feedback-value "Valid"
```

- Pour plus de détails sur l'API, reportez-vous UpdateAuthEventFeedbackà la section Référence des AWS CLI commandes.

`update-device-status`

L'exemple de code suivant montre comment utiliser`update-device-status`.

AWS CLI

Pour mettre à jour l'état de l'appareil

Cet exemple met à jour le statut d'un appareil sur « not_remembered ».

Commande :

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-
key DEVICE_KEY --device-remembered-status "not_remembered"
```

- Pour plus de détails sur l'API, reportez-vous UpdateDeviceStatusà la section Référence des AWS CLI commandes.

`update-group`

L'exemple de code suivant montre comment utiliser`update-group`.

AWS CLI

Pour mettre à jour un groupe

Cet exemple met à jour la description et la priorité de MyGroup.

Commande :

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
 --description "New description" --precedence 2
```

Sortie :

```
{
  "Group": {
      "GroupName": "MyGroup",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New description",
      "RoleArn": "arn:aws:iam::111111111111:role/MyRole",
      "Precedence": 2,
      "LastModifiedDate": 1548800862.812,
      "CreationDate": 1548097827.125
  }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateGroupà la section Référence des AWS CLI commandes.

`update-identity-provider`

L'exemple de code suivant montre comment utiliser`update-identity-provider`.

AWS CLI

Pour mettre à jour le fournisseur d'identité d'un groupe d'utilisateurs

L'`update-identity-provider`exemple suivant met à jour le fournisseur OIDC « My OIDCId P » dans le groupe d'utilisateurs demandé.

```
aws cognito-idp update-identity-provider \
    --cli-input-json file://update-identity-provider.json
```

Contenu de `update-identity-provider.json` :

```
{
   "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified",
            "username": "sub"
    },
    "CreationDate": 1.701129701653E9,
    "IdpIdentifiers": [
        "corp",
        "dev"
    ],
    "LastModifiedDate": 1.701129701653E9,
    "ProviderDetails": {
        "attributes_request_method": "GET",
        "attributes_url": "https://example.com/userInfo",
        "attributes_url_add_attributes": "false",
        "authorize_scopes": "openid profile",
        "authorize_url": "https://example.com/authorize",
        "client_id": "idpexampleclient123",
        "client_secret": "idpexamplesecret456",
        "jwks_uri": "https://example.com/.well-known/jwks.json",
        "oidc_issuer": "https://example.com",
        "token_url": "https://example.com/token"
    },
    "ProviderName": "MyOIDCIdP",
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

Sortie :

```
{
    "IdentityProvider": {
        "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified",
            "username": "sub"
        },
```

```
        "CreationDate": 1701129701.653,
        "IdpIdentifiers": [
            "corp",
            "dev"
        ],
        "LastModifiedDate": 1736444278.211,
        "ProviderDetails": {
            "attributes_request_method": "GET",
            "attributes_url": "https://example.com/userInfo",
            "attributes_url_add_attributes": "false",
            "authorize_scopes": "openid profile",
            "authorize_url": "https://example.com/authorize",
            "client_id": "idpexampleclient123",
            "client_secret": "idpexamplesecret456",
            "jwks_uri": "https://example.com/.well-known/jwks.json",
            "oidc_issuer": "https://example.com",
            "token_url": "https://example.com/token"
        },
        "ProviderName": "MyOIDCIdP",
        "ProviderType": "OIDC",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Configuration d'un domaine dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateIdentityProviderà la section Référence des AWS CLI commandes.

`update-managed-login-branding`

L'exemple de code suivant montre comment utiliser`update-managed-login-branding`.

AWS CLI

Pour mettre à jour un style de marque de connexion géré

L'`update-managed-login-branding`exemple suivant met à jour le style de marque du client d'application demandé.

```
aws cognito-idp update-managed-login-branding \
```

```
    --cli-input-json file://update-managed-login-branding.json
```

Contenu de `update-managed-login-branding.json` :

```
{
    "Assets": [
        {
            "Bytes":
 "PHN2ZyB3aWR0aD0iMjAwMDAiIGhlaWdodD0iNDAwIiB2aWV3Qm94PSIwIDAgMjAwMDAgNDAwIiBmaWxsPSJub25lIi
+CjxyZWN0IHdpZHRoPSIyMDAwMCIgaGVpZ2h0PSI0MDAiIGZpbGw9InVybCgjcGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY
+CjxsaW5lYXJHcmFkaWVudCBpZD0icGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY2NzQiIHgxPSItODk0LjI0OSIgeTE9IjE
+Cjwvc3ZnPgo=",
            "Category": "PAGE_FOOTER_BACKGROUND",
            "ColorMode": "DARK",
            "Extension": "SVG"
        }
    ],
    "ManagedLoginBrandingId": "63f30090-6b1f-4278-b885-2bbb81f8e545",
    "Settings": {
        "categories": {
            "auth": {
                "authMethodOrder": [
                    [
                        {
                            "display": "BUTTON",
                            "type": "FEDERATED"
                        },
                        {
                            "display": "INPUT",
                            "type": "USERNAME_PASSWORD"
                        }
                    ]
                ],
                "federation": {
                    "interfaceStyle": "BUTTON_LIST",
                    "order": [
                    ]
                }
            },
            "form": {
                "displayGraphics": true,
                "instructions": {
                    "enabled": false
```

```
                },
                "languageSelector": {
                    "enabled": false
                },
                "location": {
                    "horizontal": "CENTER",
                    "vertical": "CENTER"
                },
                "sessionTimerDisplay": "NONE"
            },
            "global": {
                "colorSchemeMode": "LIGHT",
                "pageFooter": {
                    "enabled": false
                },
                "pageHeader": {
                    "enabled": false
                },
                "spacingDensity": "REGULAR"
            },
            "signUp": {
                "acceptanceElements": [
                    {
                        "enforcement": "NONE",
                        "textKey": "en"
                    }
                ]
            }
        },
        "componentClasses": {
            "buttons": {
                "borderRadius": 8.0
            },
            "divider": {
                "darkMode": {
                    "borderColor": "232b37ff"
                },
                "lightMode": {
                    "borderColor": "ebebf0ff"
                }
            },
            "dropDown": {
                "borderRadius": 8.0,
                "darkMode": {
```

```
                "defaults": {
                    "itemBackgroundColor": "192534ff"
                },
                "hover": {
                    "itemBackgroundColor": "081120ff",
                    "itemBorderColor": "5f6b7aff",
                    "itemTextColor": "e9ebedff"
                },
                "match": {
                    "itemBackgroundColor": "d1d5dbff",
                    "itemTextColor": "89bdeeff"
                }
            },
            "lightMode": {
                "defaults": {
                    "itemBackgroundColor": "ffffffff"
                },
                "hover": {
                    "itemBackgroundColor": "f4f4f4ff",
                    "itemBorderColor": "7d8998ff",
                    "itemTextColor": "000716ff"
                },
                "match": {
                    "itemBackgroundColor": "414d5cff",
                    "itemTextColor": "0972d3ff"
                }
            }
        },
        "focusState": {
            "darkMode": {
                "borderColor": "539fe5ff"
            },
            "lightMode": {
                "borderColor": "0972d3ff"
            }
        },
        "idpButtons": {
            "icons": {
                "enabled": true
            }
        },
        "input": {
            "borderRadius": 8.0,
            "darkMode": {
```

```
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "5f6b7aff"
                    },
                    "placeholderColor": "8d99a8ff"
                },
                "lightMode": {
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
                    },
                    "placeholderColor": "5f6b7aff"
                }
            },
            "inputDescription": {
                "darkMode": {
                    "textColor": "8d99a8ff"
                },
                "lightMode": {
                    "textColor": "5f6b7aff"
                }
            },
            "inputLabel": {
                "darkMode": {
                    "textColor": "d1d5dbff"
                },
                "lightMode": {
                    "textColor": "000716ff"
                }
            },
            "link": {
                "darkMode": {
                    "defaults": {
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "textColor": "0972d3ff"
                    },
                    "hover": {
```

```
                    "textColor": "033160ff"
                }
            }
        },
        "optionControls": {
            "darkMode": {
                "defaults": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "7d8998ff"
                },
                "selected": {
                    "backgroundColor": "539fe5ff",
                    "foregroundColor": "000716ff"
                }
            },
            "lightMode": {
                "defaults": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "7d8998ff"
                },
                "selected": {
                    "backgroundColor": "0972d3ff",
                    "foregroundColor": "ffffffff"
                }
            }
        },
        "statusIndicator": {
            "darkMode": {
                "error": {
                    "backgroundColor": "1a0000ff",
                    "borderColor": "eb6f6fff",
                    "indicatorColor": "eb6f6fff"
                },
                "pending": {
                    "indicatorColor": "AAAAAAAA"
                },
                "success": {
                    "backgroundColor": "001a02ff",
                    "borderColor": "29ad32ff",
                    "indicatorColor": "29ad32ff"
                },
                "warning": {
                    "backgroundColor": "1d1906ff",
                    "borderColor": "e0ca57ff",
```

```
                    "indicatorColor": "e0ca57ff"
                }
            },
            "lightMode": {
                "error": {
                    "backgroundColor": "fff7f7ff",
                    "borderColor": "d91515ff",
                    "indicatorColor": "d91515ff"
                },
                "pending": {
                    "indicatorColor": "AAAAAAAA"
                },
                "success": {
                    "backgroundColor": "f2fcf3ff",
                    "borderColor": "037f0cff",
                    "indicatorColor": "037f0cff"
                },
                "warning": {
                    "backgroundColor": "fffce9ff",
                    "borderColor": "8d6605ff",
                    "indicatorColor": "8d6605ff"
                }
            }
        }
    },
    "components": {
        "alert": {
            "borderRadius": 12.0,
            "darkMode": {
                "error": {
                    "backgroundColor": "1a0000ff",
                    "borderColor": "eb6f6fff"
                }
            },
            "lightMode": {
                "error": {
                    "backgroundColor": "fff7f7ff",
                    "borderColor": "d91515ff"
                }
            }
        },
        "favicon": {
            "enabledTypes": [
                "ICO",
```

```
                "SVG"
            ]
        },
        "form": {
            "backgroundImage": {
                "enabled": false
            },
            "borderRadius": 8.0,
            "darkMode": {
                "backgroundColor": "0f1b2aff",
                "borderColor": "424650ff"
            },
            "lightMode": {
                "backgroundColor": "ffffffff",
                "borderColor": "c6c6cdff"
            },
            "logo": {
                "enabled": false,
                "formInclusion": "IN",
                "location": "CENTER",
                "position": "TOP"
            }
        },
        "idpButton": {
            "custom": {
            },
            "standard": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "354150ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    },
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "c6c6cdff",
                        "textColor": "c6c6cdff"
                    },
                    "hover": {
                        "backgroundColor": "192534ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    }
                },
```

```
                    "lightMode": {
                        "active": {
                            "backgroundColor": "d3e7f9ff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        },
                        "defaults": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "424650ff",
                            "textColor": "424650ff"
                        },
                        "hover": {
                            "backgroundColor": "f2f8fdff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        }
                    }
                }
            },
            "pageBackground": {
                "darkMode": {
                    "color": "0f1b2aff"
                },
                "image": {
                    "enabled": true
                },
                "lightMode": {
                    "color": "ffffffff"
                }
            },
            "pageFooter": {
                "backgroundImage": {
                    "enabled": false
                },
                "darkMode": {
                    "background": {
                        "color": "0f141aff"
                    },
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "background": {
                        "color": "fafafaff"
                    },
```

```
                "borderColor": "d5dbdbff"
            },
            "logo": {
                "enabled": false,
                "location": "START"
            }
        },
        "pageHeader": {
            "backgroundImage": {
                "enabled": false
            },
            "darkMode": {
                "background": {
                    "color": "0f141aff"
                },
                "borderColor": "424650ff"
            },
            "lightMode": {
                "background": {
                    "color": "fafafaff"
                },
                "borderColor": "d5dbdbff"
            },
            "logo": {
                "enabled": false,
                "location": "START"
            }
        },
        "pageText": {
            "darkMode": {
                "bodyColor": "b6bec9ff",
                "descriptionColor": "b6bec9ff",
                "headingColor": "d1d5dbff"
            },
            "lightMode": {
                "bodyColor": "414d5cff",
                "descriptionColor": "414d5cff",
                "headingColor": "000716ff"
            }
        },
        "phoneNumberSelector": {
            "displayType": "TEXT"
        },
        "primaryButton": {
```

```
                "darkMode": {
                    "active": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "defaults": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "89bdeeff",
                        "textColor": "000716ff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    },
                    "defaults": {
                        "backgroundColor": "0972d3ff",
                        "textColor": "ffffffff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    }
                }
            },
            "secondaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "354150ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    },
```

```
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "539fe5ff",
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "backgroundColor": "192534ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "0972d3ff",
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        },
        "UseCognitoProvidedValues": false,
        "UserPoolId": "ca-central-1_EXAMPLE"
}
```

Sortie :

```
{
    "ManagedLoginBranding": {
        "Assets": [
            {
                "Bytes":
 "PHN2ZyB3aWR0aD0iMjAwMDAiIGhlaWdodD0iNDAwIiB2aWV3Qm94PSIwIDAgMjAwMDAgNDAwIiBmaWxsPSJub25lIi
```

```
+CjxyZWN0IHdpZHRoPSIyMDAwMCIgaGVpZ2h0PSI0MDAiIGZpbGw9InVybCgjcGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY
+CjxsaW5lYXJHcmFkaWVudCBpZD0icGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY2NzQiIHgxPSItODk0LjI0OSIgeTE9IjE
+Cjwvc3ZnPgo=",
                "Category": "PAGE_FOOTER_BACKGROUND",
                "ColorMode": "DARK",
                "Extension": "SVG"
            }
        ],
        "CreationDate": 1732138490.642,
        "LastModifiedDate": 1732140420.301,
        "ManagedLoginBrandingId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Settings": {
            "categories": {
                "auth": {
                    "authMethodOrder": [
                        [
                            {
                                "display": "BUTTON",
                                "type": "FEDERATED"
                            },
                            {
                                "display": "INPUT",
                                "type": "USERNAME_PASSWORD"
                            }
                        ]
                    ],
                    "federation": {
                        "interfaceStyle": "BUTTON_LIST",
                        "order": [
                        ]
                    }
                },
                "form": {
                    "displayGraphics": true,
                    "instructions": {
                        "enabled": false
                    },
                    "languageSelector": {
                        "enabled": false
                    },
                    "location": {
                        "horizontal": "CENTER",
                        "vertical": "CENTER"
                    },
```

```
                "sessionTimerDisplay": "NONE"
            },
            "global": {
                "colorSchemeMode": "LIGHT",
                "pageFooter": {
                    "enabled": false
                },
                "pageHeader": {
                    "enabled": false
                },
                "spacingDensity": "REGULAR"
            },
            "signUp": {
                "acceptanceElements": [
                    {
                        "enforcement": "NONE",
                        "textKey": "en"
                    }
                ]
            }
        },
        "componentClasses": {
            "buttons": {
                "borderRadius": 8.0
            },
            "divider": {
                "darkMode": {
                    "borderColor": "232b37ff"
                },
                "lightMode": {
                    "borderColor": "ebebf0ff"
                }
            },
            "dropDown": {
                "borderRadius": 8.0,
                "darkMode": {
                    "defaults": {
                        "itemBackgroundColor": "192534ff"
                    },
                    "hover": {
                        "itemBackgroundColor": "081120ff",
                        "itemBorderColor": "5f6b7aff",
                        "itemTextColor": "e9ebedff"
                    },
```

```
                    "match": {
                        "itemBackgroundColor": "d1d5dbff",
                        "itemTextColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "itemBackgroundColor": "ffffffff"
                    },
                    "hover": {
                        "itemBackgroundColor": "f4f4f4ff",
                        "itemBorderColor": "7d8998ff",
                        "itemTextColor": "000716ff"
                    },
                    "match": {
                        "itemBackgroundColor": "414d5cff",
                        "itemTextColor": "0972d3ff"
                    }
                }
            },
            "focusState": {
                "darkMode": {
                    "borderColor": "539fe5ff"
                },
                "lightMode": {
                    "borderColor": "0972d3ff"
                }
            },
            "idpButtons": {
                "icons": {
                    "enabled": true
                }
            },
            "input": {
                "borderRadius": 8.0,
                "darkMode": {
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "5f6b7aff"
                    },
                    "placeholderColor": "8d99a8ff"
                },
                "lightMode": {
                    "defaults": {
```

```
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
                    },
                    "placeholderColor": "5f6b7aff"
                }
            },
            "inputDescription": {
                "darkMode": {
                    "textColor": "8d99a8ff"
                },
                "lightMode": {
                    "textColor": "5f6b7aff"
                }
            },
            "inputLabel": {
                "darkMode": {
                    "textColor": "d1d5dbff"
                },
                "lightMode": {
                    "textColor": "000716ff"
                }
            },
            "link": {
                "darkMode": {
                    "defaults": {
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "textColor": "033160ff"
                    }
                }
            },
            "optionControls": {
                "darkMode": {
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
```

```
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
                        "backgroundColor": "539fe5ff",
                        "foregroundColor": "000716ff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
                        "backgroundColor": "0972d3ff",
                        "foregroundColor": "ffffffff"
                    }
                }
            },
            "statusIndicator": {
                "darkMode": {
                    "error": {
                        "backgroundColor": "1a0000ff",
                        "borderColor": "eb6f6fff",
                        "indicatorColor": "eb6f6fff"
                    },
                    "pending": {
                        "indicatorColor": "AAAAAAAA"
                    },
                    "success": {
                        "backgroundColor": "001a02ff",
                        "borderColor": "29ad32ff",
                        "indicatorColor": "29ad32ff"
                    },
                    "warning": {
                        "backgroundColor": "1d1906ff",
                        "borderColor": "e0ca57ff",
                        "indicatorColor": "e0ca57ff"
                    }
                },
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff",
                        "indicatorColor": "d91515ff"
```

```
                    },
                    "pending": {
                        "indicatorColor": "AAAAAAAA"
                    },
                    "success": {
                        "backgroundColor": "f2fcf3ff",
                        "borderColor": "037f0cff",
                        "indicatorColor": "037f0cff"
                    },
                    "warning": {
                        "backgroundColor": "fffce9ff",
                        "borderColor": "8d6605ff",
                        "indicatorColor": "8d6605ff"
                    }
                }
            }
        },
        "components": {
            "alert": {
                "borderRadius": 12.0,
                "darkMode": {
                    "error": {
                        "backgroundColor": "1a0000ff",
                        "borderColor": "eb6f6fff"
                    }
                },
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff"
                    }
                }
            },
            "favicon": {
                "enabledTypes": [
                    "ICO",
                    "SVG"
                ]
            },
            "form": {
                "backgroundImage": {
                    "enabled": false
                },
                "borderRadius": 8.0,
```

```
                    "darkMode": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "424650ff"
                    },
                    "lightMode": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "c6c6cdff"
                    },
                    "logo": {
                        "enabled": false,
                        "formInclusion": "IN",
                        "location": "CENTER",
                        "position": "TOP"
                    }
                },
                "idpButton": {
                    "custom": {
                    },
                    "standard": {
                        "darkMode": {
                            "active": {
                                "backgroundColor": "354150ff",
                                "borderColor": "89bdeeff",
                                "textColor": "89bdeeff"
                            },
                            "defaults": {
                                "backgroundColor": "0f1b2aff",
                                "borderColor": "c6c6cdff",
                                "textColor": "c6c6cdff"
                            },
                            "hover": {
                                "backgroundColor": "192534ff",
                                "borderColor": "89bdeeff",
                                "textColor": "89bdeeff"
                            }
                        },
                        "lightMode": {
                            "active": {
                                "backgroundColor": "d3e7f9ff",
                                "borderColor": "033160ff",
                                "textColor": "033160ff"
                            },
                            "defaults": {
                                "backgroundColor": "ffffffff",
```

```
                        "borderColor": "424650ff",
                        "textColor": "424650ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        },
        "pageBackground": {
            "darkMode": {
                "color": "0f1b2aff"
            },
            "image": {
                "enabled": true
            },
            "lightMode": {
                "color": "ffffffff"
            }
        },
        "pageFooter": {
            "backgroundImage": {
                "enabled": false
            },
            "darkMode": {
                "background": {
                    "color": "0f141aff"
                },
                "borderColor": "424650ff"
            },
            "lightMode": {
                "background": {
                    "color": "fafafaff"
                },
                "borderColor": "d5dbdbff"
            },
            "logo": {
                "enabled": false,
                "location": "START"
            }
        },
        "pageHeader": {
```

```
            "backgroundImage": {
                "enabled": false
            },
            "darkMode": {
                "background": {
                    "color": "0f141aff"
                },
                "borderColor": "424650ff"
            },
            "lightMode": {
                "background": {
                    "color": "fafafaff"
                },
                "borderColor": "d5dbdbff"
            },
            "logo": {
                "enabled": false,
                "location": "START"
            }
        },
        "pageText": {
            "darkMode": {
                "bodyColor": "b6bec9ff",
                "descriptionColor": "b6bec9ff",
                "headingColor": "d1d5dbff"
            },
            "lightMode": {
                "bodyColor": "414d5cff",
                "descriptionColor": "414d5cff",
                "headingColor": "000716ff"
            }
        },
        "phoneNumberSelector": {
            "displayType": "TEXT"
        },
        "primaryButton": {
            "darkMode": {
                "active": {
                    "backgroundColor": "539fe5ff",
                    "textColor": "000716ff"
                },
                "defaults": {
                    "backgroundColor": "539fe5ff",
                    "textColor": "000716ff"
```

```
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "89bdeeff",
                        "textColor": "000716ff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    },
                    "defaults": {
                        "backgroundColor": "0972d3ff",
                        "textColor": "ffffffff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    }
                }
            },
            "secondaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "354150ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    },
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "539fe5ff",
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "backgroundColor": "192534ff",
                        "borderColor": "89bdeeff",
```

```
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "0972d3ff",
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        },
        "UseCognitoProvidedValues": false,
        "UserPoolId": "ca-central-1_EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Appliquer la marque aux pages de connexion gérées dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateManagedLoginBrandingà la section Référence des AWS CLI commandes.

`update-resource-server`

L'exemple de code suivant montre comment utiliser`update-resource-server`.

AWS CLI

Pour mettre à jour un serveur de ressources

Cet exemple met à jour le serveur de ressources Weather. Cela ajoute une nouvelle portée.

Commande :

```
aws cognito-idp update-resource-server --user-pool-id us-west-2_aaaaaaaaa
 --identifier weather.example.com --name Weather --scopes
 ScopeName=NewScope,ScopeDescription="New scope description"
```

Sortie :

```
{
  "ResourceServer": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Identifier": "weather.example.com",
      "Name": "Happy",
      "Scopes": [
          {
              "ScopeName": "NewScope",
              "ScopeDescription": "New scope description"
          }
      ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateResourceServerà la section Référence des AWS CLI commandes.

`update-user-attributes`

L'exemple de code suivant montre comment utiliser`update-user-attributes`.

AWS CLI

Pour mettre à jour les attributs utilisateur

Cet exemple met à jour l'attribut utilisateur « surnom ».

Commande :

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
 Name="nickname",Value="Dan"
```

- Pour plus de détails sur l'API, reportez-vous UpdateUserAttributesà la section Référence des AWS CLI commandes.

`update-user-pool-client`

L'exemple de code suivant montre comment utiliser`update-user-pool-client`.

AWS CLI

Pour mettre à jour un client d'application

L'`update-user-pool-client`exemple suivant met à jour la configuration du client d'application demandé.

```
aws cognito-idp update-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --client-name my-test-app \
    --refresh-token-validity 30 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days \
    --read-
attributes "address" "birthdate" "email" "email_verified" "family_name" "gender" "locale" "m
 \
    --write-
attributes "address" "birthdate" "email" "family_name" "gender" "locale" "middle_name" "name
 \
    --explicit-auth-
flows "ALLOW_ADMIN_USER_PASSWORD_AUTH" "ALLOW_CUSTOM_AUTH" "ALLOW_REFRESH_TOKEN_AUTH" "ALLOW
 \
    --supported-identity-providers "MySAML" "COGNITO" "Google" \
    --callback-urls "https://www.example.com" "https://app2.example.com" \
    --logout-urls "https://auth.example.com/login?
client_id=1example23456789&response_type=code&redirect_uri=https%3A%2F
%2Fwww.example.com" "https://example.com/logout" \
    --default-redirect-uri "https://www.example.com" \
    --allowed-o-auth-flows "code" "implicit" \
    --allowed-o-auth-scopes "openid" "profile" "aws.cognito.signin.user.admin" \
    --allowed-o-auth-flows-user-pool-client \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --no-enable-propagate-additional-user-context-data \
    --auth-session-validity 3
```

Sortie :

```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "my-test-app",
        "ClientId": "1example23456789",
        "LastModifiedDate": "2025-01-31T14:40:12.498000-08:00",
        "CreationDate": "2023-09-13T16:26:34.408000-07:00",
        "RefreshTokenValidity": 30,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "website",
            "zoneinfo",
            "address",
            "birthdate",
            "email_verified",
            "gender",
            "profile",
            "phone_number_verified",
            "preferred_username",
            "locale",
            "middle_name",
            "picture",
            "updated_at",
            "name",
            "nickname",
            "phone_number",
            "family_name",
            "email"
        ],
        "WriteAttributes": [
            "website",
            "zoneinfo",
            "address",
            "birthdate",
            "gender",
            "profile",
            "preferred_username",
```

```
            "locale",
            "middle_name",
            "picture",
            "updated_at",
            "name",
            "nickname",
            "phone_number",
            "family_name",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_CUSTOM_AUTH",
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_ADMIN_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "COGNITO",
            "MySAML"
        ],
        "CallbackURLs": [
            "https://www.example.com",
            "https://app2.example.com"
        ],
        "LogoutURLs": [
            "https://example.com/logout",
            "https://auth.example.com/login?
client_id=1example23456789&response_type=code&redirect_uri=https%3A%2F
%2Fwww.example.com"
        ],
        "DefaultRedirectURI": "https://www.example.com",
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "PreventUserExistenceErrors": "ENABLED",
```

```
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": false,
        "AuthSessionValidity": 3
    }
}
```

Pour plus d'informations, consultez la section Paramètres spécifiques à l'application avec les clients d'applications dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateUserPoolClientà la section Référence des AWS CLI commandes.

`update-user-pool-domain`

L'exemple de code suivant montre comment utiliser`update-user-pool-domain`.

AWS CLI

Pour mettre à jour un domaine personnalisé

L'`update-user-pool-domain`exemple suivant configure la version de marque et le certificat pour le domaine personnalisé du groupe d'utilisateurs demandé.

```
aws cognito-idp update-user-pool-domain \
    --user-pool-id ca-central-1_EXAMPLE \
    --domain auth.example.com \
    --managed-login-version 2 \
    --custom-domain-config CertificateArn=arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "CloudFrontDomain": "example.cloudfront.net",
    "ManagedLoginVersion": 2
}
```

Pour plus d'informations, consultez les sections Connexion gérée et Configuration d'un domaine dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateUserPoolDomainà la section Référence des AWS CLI commandes.

update-user-pool

L'exemple de code suivant montre comment utiliserupdate-user-pool.

AWS CLI

Pour mettre à jour un groupe d'utilisateurs

L'update-user-poolexemple suivant modifie un groupe d'utilisateurs avec un exemple de syntaxe pour chacune des options de configuration disponibles. Pour mettre à jour un groupe d'utilisateurs, vous devez spécifier toutes les options précédemment configurées, sinon elles seront réinitialisées à une valeur par défaut.

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
    --policies PasswordPolicy=
\{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,RequireSym
 \
    --deletion-protection ACTIVE \
    --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
    --auto-verified-attributes "phone_number" "email" \
    --verification-message-template \{\"SmsMessage\":\""Your code is
 {####}"\",\"EmailMessage\":\""Your code is {####}"\",\"EmailSubject\":\""Your
 verification code"\",\"EmailMessageByLink\":\""Click {##here##} to verify
 your email address."\",\"EmailSubjectByLink\":\""Your verification link"\",
\"DefaultEmailOption\":\"CONFIRM_WITH_LINK\"\} \
    --sms-authentication-message "Your code is {####}" \
    --user-attribute-update-settings
 AttributesRequireVerificationBeforeUpdate="email","phone_number" \
    --mfa-configuration "OPTIONAL" \
    --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
 \
    --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="amdin
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",ConfigurationSet
configuration-set" \
    --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-role/
SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
    --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
```

```
 {####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailSubject=\""Welcome to MyMobileGame"\"\} \
    --user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
    --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailSubject=\""Welcome to MyMobileGame"\"\} \
    --user-pool-add-ons AdvancedSecurityMode="AUDIT" \
    --account-recovery-setting RecoveryMechanisms=
\[\{Priority=1,Name="verified_email"\},\{Priority=2,Name="verified_phone_number"\}\]
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Mise à jour de la configuration du groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateUserPoolà la section Référence des AWS CLI commandes.

`verify-software-token`

L'exemple de code suivant montre comment utiliser`verify-software-token`.

AWS CLI

Pour confirmer l'enregistrement d'un authentificateur TOTP

L'`verify-software-token`exemple suivant complète l'enregistrement TOTP pour l'utilisateur actuel.

```
aws cognito-idp verify-software-token \
    --access-token eyJra456defEXAMPLE \
    --user-code 123456
```

Sortie :

```
{
    "Status": "SUCCESS"
}
```

Pour plus d'informations, consultez la section Ajouter un MFA à un groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous VerifySoftwareTokenà la section Référence des AWS CLI commandes.

`verify-user-attribute`

L'exemple de code suivant montre comment utiliser`verify-user-attribute`.

AWS CLI

Pour vérifier une modification d'attribut

L'`verify-user-attribute`exemple suivant vérifie une modification apportée à l'attribut e-mail de l'utilisateur actuel.

```
aws cognito-idp verify-user-attribute \
    --access-token eyJra456defEXAMPLE \
    --attribute-name email \
    --code 123456
```

Pour plus d'informations, consultez la section Configuration de la vérification par e-mail ou par téléphone dans le manuel Amazon Cognito Developer Guide.

- Pour plus de détails sur l'API, reportez-vous VerifyUserAttributeà la section Référence des AWS CLI commandes.

Exemples d'Amazon Comprehend utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Comprehend.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-detect-dominant-language`

L'exemple de code suivant montre comment utiliser`batch-detect-dominant-language`.

AWS CLI

Pour détecter la langue dominante de plusieurs textes d'entrée

L'`batch-detect-dominant-language`exemple suivant analyse plusieurs textes d'entrée et renvoie la langue dominante de chacun d'entre eux. Le score de confiance des modèles pré-entraînés est également généré pour chaque prédiction.

```
aws comprehend batch-detect-dominant-language \
    --text-list "Physics is the natural science that involves the study of matter
 and its motion and behavior through space and time, along with related concepts
 such as energy and force."
```

Sortie :

```
{
    "ResultList": [
        {
            "Index": 0,
            "Languages": [
                {
                    "LanguageCode": "en",
                    "Score": 0.9986501932144165
                }
            ]
        }
    ],
    "ErrorList": []
}
```

Pour plus d'informations, consultez Dominant Language dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDetectDominantLanguageà la section Référence des AWS CLI commandes.

`batch-detect-entities`

L'exemple de code suivant montre comment utiliser`batch-detect-entities`.

AWS CLI

Pour détecter des entités à partir de plusieurs textes d'entrée

L'`batch-detect-entities`exemple suivant analyse plusieurs textes d'entrée et renvoie les entités nommées de chacun d'entre eux. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend batch-detect-entities \
    --language-code en \
    --text-list "Dear Jane, Your AnyCompany Financial Services LLC credit card account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July 31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "ResultList": [
        {
            "Index": 0,
            "Entities": [
                {
                    "Score": 0.9985517859458923,
                    "Type": "PERSON",
                    "Text": "Jane",
                    "BeginOffset": 5,
                    "EndOffset": 9
                },
                {
                    "Score": 0.9767839312553406,
                    "Type": "ORGANIZATION",
                    "Text": "AnyCompany Financial Services, LLC",
                    "BeginOffset": 16,
                    "EndOffset": 50
                },
                {
                    "Score": 0.9856694936752319,
                    "Type": "OTHER",
```

```
                    "Text": "1111-XXXX-1111-XXXX",
                    "BeginOffset": 71,
                    "EndOffset": 90
                },
                {
                    "Score": 0.9652159810066223,
                    "Type": "QUANTITY",
                    "Text": ".53",
                    "BeginOffset": 116,
                    "EndOffset": 119
                },
                {
                    "Score": 0.9986667037010193,
                    "Type": "DATE",
                    "Text": "July 31st",
                    "BeginOffset": 135,
                    "EndOffset": 144
                }
            ]
        },
        {
            "Index": 1,
            "Entities": [
                {
                    "Score": 0.720084547996521,
                    "Type": "ORGANIZATION",
                    "Text": "Sunshine Spa",
                    "BeginOffset": 33,
                    "EndOffset": 45
                },
                {
                    "Score": 0.9865870475769043,
                    "Type": "LOCATION",
                    "Text": "123 Main St",
                    "BeginOffset": 47,
                    "EndOffset": 58
                },
                {
                    "Score": 0.5895616412162781,
                    "Type": "LOCATION",
                    "Text": "Anywhere",
                    "BeginOffset": 60,
                    "EndOffset": 68
                },
```

```
                {
                    "Score": 0.6809214353561401,
                    "Type": "PERSON",
                    "Text": "Alice",
                    "BeginOffset": 75,
                    "EndOffset": 80
                },
                {
                    "Score": 0.9979087114334106,
                    "Type": "OTHER",
                    "Text": "AnySpa@example.com",
                    "BeginOffset": 84,
                    "EndOffset": 99
                }
            ]
        }
    ],
    "ErrorList": []
}
```

Pour plus d'informations, consultez Entities dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDetectEntitiesà la section Référence des AWS CLI commandes.

`batch-detect-key-phrases`

L'exemple de code suivant montre comment utiliser`batch-detect-key-phrases`.

AWS CLI

Pour détecter les phrases clés de plusieurs entrées de texte

L'`batch-detect-key-phrases`exemple suivant analyse plusieurs textes d'entrée et renvoie les phrases nominales clés de chacun d'entre eux. Le score de confiance du modèle préentraîné pour chaque prédiction est également généré.

```
aws comprehend batch-detect-key-phrases \
    --language-code en \
    --text-list "Hello Zhang Wei, I am John, writing to you about the trip for
 next Saturday." "Dear Jane, Your AnyCompany Financial Services LLC credit card
```

```
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to
Alice at AnySpa@example.com."
```

Sortie :

```
{
    "ResultList": [
        {
            "Index": 0,
            "KeyPhrases": [
                {
                    "Score": 0.99700927734375,
                    "Text": "Zhang Wei",
                    "BeginOffset": 6,
                    "EndOffset": 15
                },
                {
                    "Score": 0.9929308891296387,
                    "Text": "John",
                    "BeginOffset": 22,
                    "EndOffset": 26
                },
                {
                    "Score": 0.9997230172157288,
                    "Text": "the trip",
                    "BeginOffset": 49,
                    "EndOffset": 57
                },
                {
                    "Score": 0.9999470114707947,
                    "Text": "next Saturday",
                    "BeginOffset": 62,
                    "EndOffset": 75
                }
            ]
        },
        {
            "Index": 1,
            "KeyPhrases": [
                {
                    "Score": 0.8358274102210999,
                    "Text": "Dear Jane",
```

```
                    "BeginOffset": 0,
                    "EndOffset": 9
                },
                {
                    "Score": 0.989359974861145,
                    "Text": "Your AnyCompany Financial Services",
                    "BeginOffset": 11,
                    "EndOffset": 45
                },
                {
                    "Score": 0.8812323808670044,
                    "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
                    "BeginOffset": 47,
                    "EndOffset": 90
                },
                {
                    "Score": 0.9999381899833679,
                    "Text": "a minimum payment",
                    "BeginOffset": 95,
                    "EndOffset": 112
                },
                {
                    "Score": 0.9997439980506897,
                    "Text": ".53",
                    "BeginOffset": 116,
                    "EndOffset": 119
                },
                {
                    "Score": 0.996875524520874,
                    "Text": "July 31st",
                    "BeginOffset": 135,
                    "EndOffset": 144
                }
            ]
        },
        {
            "Index": 2,
            "KeyPhrases": [
                {
                    "Score": 0.9990295767784119,
                    "Text": "customer feedback",
                    "BeginOffset": 12,
                    "EndOffset": 29
                },
```

```
                {
                    "Score": 0.9994127750396729,
                    "Text": "Sunshine Spa",
                    "BeginOffset": 33,
                    "EndOffset": 45
                },
                {
                    "Score": 0.9892991185188293,
                    "Text": "123 Main St",
                    "BeginOffset": 47,
                    "EndOffset": 58
                },
                {
                    "Score": 0.9969810843467712,
                    "Text": "Alice",
                    "BeginOffset": 75,
                    "EndOffset": 80
                },
                {
                    "Score": 0.9703696370124817,
                    "Text": "AnySpa@example.com",
                    "BeginOffset": 84,
                    "EndOffset": 99
                }
            ]
        }
    ],
    "ErrorList": []
}
```

Pour plus d'informations, consultez les phrases clés du manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDetectKeyPhrasesà la section Référence des AWS CLI commandes.

`batch-detect-sentiment`

L'exemple de code suivant montre comment utiliser`batch-detect-sentiment`.

AWS CLI

Pour détecter le sentiment qui prévaut lors de plusieurs saisies de textes

L'`batch-detect-sentiment`exemple suivant analyse plusieurs textes saisis et renvoie le sentiment dominant (`POSITIVE`, `NEUTRAL``MIXED`, ou`NEGATIVE`, de chacun d'eux).

```
aws comprehend batch-detect-sentiment \
    --text-list "That movie was very boring, I can't believe it was over four hours long." "It is a beautiful day for hiking today." "My meal was okay, I'm excited to try other restaurants." \
    --language-code en
```

Sortie :

```
{
    "ResultList": [
        {
            "Index": 0,
            "Sentiment": "NEGATIVE",
            "SentimentScore": {
                "Positive": 0.00011316669406369328,
                "Negative": 0.9995445609092712,
                "Neutral": 0.00014722718333359808,
                "Mixed": 0.00019498742767609656
            }
        },
        {
            "Index": 1,
            "Sentiment": "POSITIVE",
            "SentimentScore": {
                "Positive": 0.9981263279914856,
                "Negative": 0.00015240783977787942,
                "Neutral": 0.0013876151060685515,
                "Mixed": 0.00033366199932061136
            }
        },
        {
            "Index": 2,
            "Sentiment": "MIXED",
            "SentimentScore": {
                "Positive": 0.15930435061454773,
                "Negative": 0.11471917480230331,
                "Neutral": 0.26897063851356506,
                "Mixed": 0.45700588822364807
            }
        }
```

```
    ],
    "ErrorList": []
}
```

Pour plus d'informations, consultez Sentiment dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDetectSentimentà la section Référence des AWS CLI commandes.

`batch-detect-syntax`

L'exemple de code suivant montre comment utiliser`batch-detect-syntax`.

AWS CLI

Pour inspecter la syntaxe et les parties du discours des mots dans plusieurs textes d'entrée

L'`batch-detect-syntax`exemple suivant analyse la syntaxe de plusieurs textes d'entrée et renvoie les différentes parties du discours. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend batch-detect-syntax \
    --text-list "It is a beautiful day." "Can you please pass the salt?" "Please pay
 the bill before the 31st." \
    --language-code en
```

Sortie :

```
{
    "ResultList": [
        {
            "Index": 0,
            "SyntaxTokens": [
                {
                    "TokenId": 1,
                    "Text": "It",
                    "BeginOffset": 0,
                    "EndOffset": 2,
                    "PartOfSpeech": {
                        "Tag": "PRON",
                        "Score": 0.9999740719795227
```

```
            }
        },
        {
            "TokenId": 2,
            "Text": "is",
            "BeginOffset": 3,
            "EndOffset": 5,
            "PartOfSpeech": {
                "Tag": "VERB",
                "Score": 0.999937117099762
            }
        },
        {
            "TokenId": 3,
            "Text": "a",
            "BeginOffset": 6,
            "EndOffset": 7,
            "PartOfSpeech": {
                "Tag": "DET",
                "Score": 0.9999926686286926
            }
        },
        {
            "TokenId": 4,
            "Text": "beautiful",
            "BeginOffset": 8,
            "EndOffset": 17,
            "PartOfSpeech": {
                "Tag": "ADJ",
                "Score": 0.9987891912460327
            }
        },
        {
            "TokenId": 5,
            "Text": "day",
            "BeginOffset": 18,
            "EndOffset": 21,
            "PartOfSpeech": {
                "Tag": "NOUN",
                "Score": 0.9999778866767883
            }
        },
        {
            "TokenId": 6,
```

```
                    "Text": ".",
                    "BeginOffset": 21,
                    "EndOffset": 22,
                    "PartOfSpeech": {
                        "Tag": "PUNCT",
                        "Score": 0.9999974966049194
                    }
                }
            ]
        },
        {
            "Index": 1,
            "SyntaxTokens": [
                {
                    "TokenId": 1,
                    "Text": "Can",
                    "BeginOffset": 0,
                    "EndOffset": 3,
                    "PartOfSpeech": {
                        "Tag": "AUX",
                        "Score": 0.9999770522117615
                    }
                },
                {
                    "TokenId": 2,
                    "Text": "you",
                    "BeginOffset": 4,
                    "EndOffset": 7,
                    "PartOfSpeech": {
                        "Tag": "PRON",
                        "Score": 0.9999986886978149
                    }
                },
                {
                    "TokenId": 3,
                    "Text": "please",
                    "BeginOffset": 8,
                    "EndOffset": 14,
                    "PartOfSpeech": {
                        "Tag": "INTJ",
                        "Score": 0.9681622385978699
                    }
                },
                {
```

```
                    "TokenId": 4,
                    "Text": "pass",
                    "BeginOffset": 15,
                    "EndOffset": 19,
                    "PartOfSpeech": {
                        "Tag": "VERB",
                        "Score": 0.9999874830245972
                    }
                },
                {
                    "TokenId": 5,
                    "Text": "the",
                    "BeginOffset": 20,
                    "EndOffset": 23,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999827146530151
                    }
                },
                {
                    "TokenId": 6,
                    "Text": "salt",
                    "BeginOffset": 24,
                    "EndOffset": 28,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
                        "Score": 0.9995040893554688
                    }
                },
                {
                    "TokenId": 7,
                    "Text": "?",
                    "BeginOffset": 28,
                    "EndOffset": 29,
                    "PartOfSpeech": {
                        "Tag": "PUNCT",
                        "Score": 0.999998152256012
                    }
                }
            ]
        },
        {
            "Index": 2,
            "SyntaxTokens": [
```

```
                {
                    "TokenId": 1,
                    "Text": "Please",
                    "BeginOffset": 0,
                    "EndOffset": 6,
                    "PartOfSpeech": {
                        "Tag": "INTJ",
                        "Score": 0.9997857809066772
                    }
                },
                {
                    "TokenId": 2,
                    "Text": "pay",
                    "BeginOffset": 7,
                    "EndOffset": 10,
                    "PartOfSpeech": {
                        "Tag": "VERB",
                        "Score": 0.9999252557754517
                    }
                },
                {
                    "TokenId": 3,
                    "Text": "the",
                    "BeginOffset": 11,
                    "EndOffset": 14,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999842643737793
                    }
                },
                {
                    "TokenId": 4,
                    "Text": "bill",
                    "BeginOffset": 15,
                    "EndOffset": 19,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
                        "Score": 0.9999588131904602
                    }
                },
                {
                    "TokenId": 5,
                    "Text": "before",
                    "BeginOffset": 20,
```

```
                    "EndOffset": 26,
                    "PartOfSpeech": {
                        "Tag": "ADP",
                        "Score": 0.9958304762840271
                    }
                },
                {
                    "TokenId": 6,
                    "Text": "the",
                    "BeginOffset": 27,
                    "EndOffset": 30,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999947547912598
                    }
                },
                {
                    "TokenId": 7,
                    "Text": "31st",
                    "BeginOffset": 31,
                    "EndOffset": 35,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
                        "Score": 0.9924124479293823
                    }
                },
                {
                    "TokenId": 8,
                    "Text": ".",
                    "BeginOffset": 35,
                    "EndOffset": 36,
                    "PartOfSpeech": {
                        "Tag": "PUNCT",
                        "Score": 0.9999955892562866
                    }
                }
            ]
        }
    ],
    "ErrorList": []
}
```

Pour plus d'informations, consultez la section Analyse syntaxique dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDetectSyntaxà la section Référence des AWS CLI commandes.

`batch-detect-targeted-sentiment`

L'exemple de code suivant montre comment utiliser`batch-detect-targeted-sentiment`.

AWS CLI

Pour détecter le sentiment et chaque entité nommée pour plusieurs textes d'entrée

L'`batch-detect-targeted-sentiment`exemple suivant analyse plusieurs textes saisis et renvoie les entités nommées ainsi que le sentiment dominant associé à chaque entité. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend batch-detect-targeted-sentiment \
    --language-code en \
    --text-list "That movie was really boring, the original was way more
 entertaining" "The trail is extra beautiful today." "My meal was just okay."
```

Sortie :

```
{
    "ResultList": [
        {
            "Index": 0,
            "Entities": [
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9999009966850281,
                            "GroupScore": 1.0,
                            "Text": "movie",
                            "Type": "MOVIE",
                            "MentionSentiment": {
                                "Sentiment": "NEGATIVE",
                                "SentimentScore": {
```

```
                                "Positive": 0.13887299597263336,
                                "Negative": 0.8057460188865662,
                                "Neutral": 0.05525200068950653,
                                "Mixed": 0.00012799999967683107
                            }
                        },
                        "BeginOffset": 5,
                        "EndOffset": 10
                    }
                ]
            },
            {
                "DescriptiveMentionIndex": [
                    0
                ],
                "Mentions": [
                    {
                        "Score": 0.9921110272407532,
                        "GroupScore": 1.0,
                        "Text": "original",
                        "Type": "MOVIE",
                        "MentionSentiment": {
                            "Sentiment": "POSITIVE",
                            "SentimentScore": {
                                "Positive": 0.9999989867210388,
                                "Negative": 9.999999974752427e-07,
                                "Neutral": 0.0,
                                "Mixed": 0.0
                            }
                        },
                        "BeginOffset": 34,
                        "EndOffset": 42
                    }
                ]
            }
        ]
    },
    {
        "Index": 1,
        "Entities": [
            {
                "DescriptiveMentionIndex": [
                    0
                ],
```

```
            "Mentions": [
                {
                    "Score": 0.7545599937438965,
                    "GroupScore": 1.0,
                    "Text": "trail",
                    "Type": "OTHER",
                    "MentionSentiment": {
                        "Sentiment": "POSITIVE",
                        "SentimentScore": {
                            "Positive": 1.0,
                            "Negative": 0.0,
                            "Neutral": 0.0,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 4,
                    "EndOffset": 9
                }
            ]
        },
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.9999960064888,
                    "GroupScore": 1.0,
                    "Text": "today",
                    "Type": "DATE",
                    "MentionSentiment": {
                        "Sentiment": "NEUTRAL",
                        "SentimentScore": {
                            "Positive": 9.000000318337698e-06,
                            "Negative": 1.9999999949504854e-06,
                            "Neutral": 0.9999859929084778,
                            "Mixed": 3.999999989900971e-06
                        }
                    },
                    "BeginOffset": 29,
                    "EndOffset": 34
                }
            ]
        }
```

```
            ]
        },
        {
            "Index": 2,
            "Entities": [
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9999880194664001,
                            "GroupScore": 1.0,
                            "Text": "My",
                            "Type": "PERSON",
                            "MentionSentiment": {
                                "Sentiment": "NEUTRAL",
                                "SentimentScore": {
                                    "Positive": 0.0,
                                    "Negative": 0.0,
                                    "Neutral": 1.0,
                                    "Mixed": 0.0
                                }
                            },
                            "BeginOffset": 0,
                            "EndOffset": 2
                        }
                    ]
                },
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9995260238647461,
                            "GroupScore": 1.0,
                            "Text": "meal",
                            "Type": "OTHER",
                            "MentionSentiment": {
                                "Sentiment": "NEUTRAL",
                                "SentimentScore": {
                                    "Positive": 0.04695599898695946,
                                    "Negative": 0.003226999891921878,
```

```
                                "Neutral": 0.6091709733009338,
                                "Mixed": 0.34064599871635437
                            }
                        },
                        "BeginOffset": 3,
                        "EndOffset": 7
                    }
                ]
            }
        ]
    }
    ],
    "ErrorList": []
}
```

Pour plus d'informations, consultez Targeted Sentiment dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDetectTargetedSentimentà la section Référence des AWS CLI commandes.

`classify-document`

L'exemple de code suivant montre comment utiliser`classify-document`.

AWS CLI

Pour classer un document avec un point de terminaison spécifique au modèle

L'`classify-document`exemple suivant classe un document avec le point final d'un modèle personnalisé. Le modèle de cet exemple a été entraîné sur un ensemble de données contenant des messages SMS étiquetés comme spam ou non, ou « ham ».

```
aws comprehend classify-document \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint \
    --text "CONGRATULATIONS! TXT 1235550100 to win $5000"
```

Sortie :

```
{
```

```
    "Classes": [
        {
            "Name": "spam",
            "Score": 0.9998599290847778
        },
        {
            "Name": "ham",
            "Score": 0.00014001205272506922
        }
    ]
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ClassifyDocumentà la section Référence des AWS CLI commandes.

`contains-pii-entities`

L'exemple de code suivant montre comment utiliser`contains-pii-entities`.

AWS CLI

Pour analyser le texte saisi pour détecter la présence d'informations personnelles

L'`contains-pii-entities`exemple suivant analyse le texte saisi pour détecter la présence d'informations personnelles identifiables (PII) et renvoie les étiquettes des types d'entités PII identifiés tels que le nom, l'adresse, le numéro de compte bancaire ou le numéro de téléphone.

```
aws comprehend contains-pii-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card
       account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
 July 31st. Based on your autopay settings,
       we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000.
       Customer feedback for Sunshine Spa, 100 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

Sortie :

```
{
    "Labels": [
        {
            "Name": "NAME",
            "Score": 1.0
        },
        {
            "Name": "EMAIL",
            "Score": 1.0
        },
        {
            "Name": "BANK_ACCOUNT_NUMBER",
            "Score": 0.9995794296264648
        },
        {
            "Name": "BANK_ROUTING",
            "Score": 0.9173126816749573
        },
        {
            "Name": "CREDIT_DEBIT_NUMBER",
            "Score": 1.0
        }
}
```

Pour plus d'informations, consultez la section Informations personnelles identifiables (PII) dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ContainsPiiEntitiesà la section Référence des AWS CLI commandes.

`create-dataset`

L'exemple de code suivant montre comment utiliser`create-dataset`.

AWS CLI

Pour créer un jeu de données sur le volant

L'`create-dataset`exemple suivant crée un jeu de données pour un volant d'inertie. Cet ensemble de données sera utilisé comme données d'entraînement supplémentaires, comme indiqué par la `--dataset-type` balise.

```
aws comprehend create-dataset \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity \
    --dataset-name example-dataset \
    --dataset-type "TRAIN" \
    --input-data-config file://inputConfig.json
```

Contenu de `file://inputConfig.json` :

```
{
    "DataFormat": "COMPREHEND_CSV",
    "DocumentClassifierInputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-bucket/training-data.csv"
    }
}
```

Sortie :

```
{
    "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset"
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le guide du développeur Amazon Comprehend.

- Pour plus de détails sur l'API, reportez-vous CreateDatasetà la section Référence des AWS CLI commandes.

`create-document-classifier`

L'exemple de code suivant montre comment utiliser`create-document-classifier`.

AWS CLI

Pour créer un classificateur de documents afin de classer les documents

L'`create-document-classifier`exemple suivant commence le processus de formation pour un modèle de classificateur de documents. Le fichier de données d'`training.csv`entraînement se trouve dans le `--input-data-config` tag. `training.csv`est un document à deux

colonnes où les étiquettes ou les classifications sont fournies dans la première colonne et les documents sont fournis dans la deuxième colonne.

```
aws comprehend create-document-classifier \
    --document-classifier-name example-classifier \
    --data-access-arn arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/123456abcdeb0e11022f22a11EXAMPLE \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --language-code en
```

Sortie :

```
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDocumentClassifierà la section Référence des AWS CLI commandes.

`create-endpoint`

L'exemple de code suivant montre comment utiliser`create-endpoint`.

AWS CLI

Pour créer un point de terminaison pour un modèle personnalisé

L'`create-endpoint`exemple suivant crée un point de terminaison pour l'inférence synchrone pour un modèle personnalisé préalablement entraîné.

```
aws comprehend create-endpoint \
    --endpoint-name example-classifier-endpoint-1 \
    --model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier \
    --desired-inference-units 1
```

Sortie :

```
{
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint-1"
}
```

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateEndpointà la section Référence des AWS CLI commandes.

`create-entity-recognizer`

L'exemple de code suivant montre comment utiliser`create-entity-recognizer`.

AWS CLI

Pour créer un outil de reconnaissance d'entités personnalisé

L'`create-entity-recognizer`exemple suivant lance le processus de formation pour un modèle de reconnaissance d'entités personnalisé. Cet exemple utilise un fichier CSV contenant des documents de formation et une liste d'entités CSV `entity_list.csv` pour entraîner le modèle. `raw_text.csv` `entity-list.csv`contient les colonnes suivantes : texte et type.

```
aws comprehend create-entity-recognizer \
    --recognizer-name example-entity-recognizer
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --input-data-config "EntityTypes=[{Type=DEVICE}],Documents={S3Uri=s3://amzn-s3-
demo-bucket/trainingdata/raw_text.csv},EntityList={S3Uri=s3://amzn-s3-demo-bucket/
trainingdata/entity_list.csv}"
    --language-code en
```

Sortie :

```
{
    "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-
entity-recognizer/entityrecognizer1"
}
```

Pour plus d'informations, consultez la section Reconnaissance d'entités personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateEntityRecognizerà la section Référence des AWS CLI commandes.

`create-flywheel`

L'exemple de code suivant montre comment utiliser`create-flywheel`.

AWS CLI

Pour créer un volant

L'`create-flywheel`exemple suivant crée un volant pour orchestrer la formation continue d'un modèle de classification de documents ou de reconnaissance d'entités. Dans cet exemple, le volant est créé pour gérer un modèle entraîné existant spécifié par le `--active-model-arn` tag. Lorsque le volant est créé, un lac de données est créé au niveau de la `--input-data-lake` balise.

```
aws comprehend create-flywheel \
    --flywheel-name example-flywheel \
    --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-model/version/1 \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --data-lake-s3-uri "s3://amzn-s3-demo-bucket"
```

Sortie :

```
{
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel"
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le guide du développeur Amazon Comprehend.

- Pour plus de détails sur l'API, reportez-vous CreateFlywheelà la section Référence des AWS CLI commandes.

`delete-document-classifier`

L'exemple de code suivant montre comment utiliser`delete-document-classifier`.

AWS CLI

Pour supprimer un classificateur de documents personnalisé

L'`delete-document-classifier`exemple suivant supprime un modèle de classificateur de documents personnalisé.

```
aws comprehend delete-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDocumentClassifierà la section Référence des AWS CLI commandes.

`delete-endpoint`

L'exemple de code suivant montre comment utiliser`delete-endpoint`.

AWS CLI

Pour supprimer un point de terminaison pour un modèle personnalisé

L'`delete-endpoint`exemple suivant supprime un point de terminaison spécifique au modèle. Tous les points de terminaison doivent être supprimés pour que le modèle soit supprimé.

```
aws comprehend delete-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteEndpointà la section Référence des AWS CLI commandes.

`delete-entity-recognizer`

L'exemple de code suivant montre comment utiliser`delete-entity-recognizer`.

AWS CLI

Pour supprimer un modèle de reconnaissance d'entités personnalisé

L'`delete-entity-recognizer`exemple suivant supprime un modèle de reconnaissance d'entités personnalisé.

```
aws comprehend delete-entity-recognizer \
    --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/example-entity-recognizer-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteEntityRecognizerà la section Référence des AWS CLI commandes.

`delete-flywheel`

L'exemple de code suivant montre comment utiliser`delete-flywheel`.

AWS CLI

Pour supprimer un volant

L'`delete-flywheel`exemple suivant supprime un volant. Le lac de données ou le modèle associé au volant n'est pas supprimé.

```
aws comprehend delete-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la présentation de Flywheel dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteFlywheelà la section Référence des AWS CLI commandes.

`delete-resource-policy`

L'exemple de code suivant montre comment utiliser`delete-resource-policy`.

AWS CLI

Pour supprimer une politique basée sur les ressources

L'`delete-resource-policy`exemple suivant supprime une politique basée sur les ressources d'une ressource Amazon Comprehend.

```
aws comprehend delete-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier-1/version/1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Copier des modèles personnalisés entre AWS comptes dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteResourcePolicyà la section Référence des AWS CLI commandes.

`describe-dataset`

L'exemple de code suivant montre comment utiliser`describe-dataset`.

AWS CLI

Pour décrire un jeu de données sur un volant

L'`describe-dataset`exemple suivant permet d'obtenir les propriétés d'un jeu de données volant.

```
aws comprehend describe-dataset \
```

```
    --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset
```

Sortie :

```
{
    "DatasetProperties": {
        "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset",
        "DatasetName": "example-dataset",
        "DatasetType": "TRAIN",
        "DatasetS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/
schemaVersion=1/12345678A123456Z/datasets/example-dataset/20230616T203710Z/",
        "Status": "CREATING",
        "CreationTime": "2023-06-16T20:37:10.400000+00:00"
    }
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le guide du développeur Amazon Comprehend.

- Pour plus de détails sur l'API, reportez-vous DescribeDatasetà la section Référence des AWS CLI commandes.

`describe-document-classification-job`

L'exemple de code suivant montre comment utiliser`describe-document-classification-job`.

AWS CLI

Pour décrire une tâche de classification de documents

L'`describe-document-classification-job`exemple suivant obtient les propriétés d'une tâche de classification de documents asynchrone.

```
aws comprehend describe-document-classification-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "DocumentClassificationJobProperties": {
```

```
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "exampleclassificationjob",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
        "EndTime": "2023-06-14T17:15:58.582000+00:00",
        "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/mymodel/version/1",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
CLN-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDocumentClassificationJobà la section Référence des AWS CLI commandes.

`describe-document-classifier`

L'exemple de code suivant montre comment utiliser`describe-document-classifier`.

AWS CLI

Pour décrire un classificateur de documents

L'`describe-document-classifier`exemple suivant permet d'obtenir les propriétés d'un modèle de classificateur de documents personnalisé.

```
aws comprehend describe-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier-1
```

Sortie :

```
{
    "DocumentClassifierProperties": {
        "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier-1",
        "LanguageCode": "en",
        "Status": "TRAINED",
        "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
        "EndTime": "2023-06-13T19:42:31.752000+00:00",
        "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
        "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
        "InputDataConfig": {
            "DataFormat": "COMPREHEND_CSV",
            "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
        },
        "OutputDataConfig": {},
        "ClassifierMetadata": {
            "NumberOfLabels": 3,
            "NumberOfTrainedDocuments": 5016,
            "NumberOfTestDocuments": 557,
            "EvaluationMetrics": {
                "Accuracy": 0.9856,
                "Precision": 0.9919,
                "Recall": 0.9459,
                "F1Score": 0.9673,
                "MicroPrecision": 0.9856,
                "MicroRecall": 0.9856,
                "MicroF1Score": 0.9856,
                "HammingLoss": 0.0144
            }
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "Mode": "MULTI_CLASS"
    }
}
```

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDocumentClassifierà la section Référence des AWS CLI commandes.

`describe-dominant-language-detection-job`

L'exemple de code suivant montre comment utiliser`describe-dominant-language-detection-job`.

AWS CLI

Décrire une tâche de détection du langage dominant.

L'`describe-dominant-language-detection-job`exemple suivant obtient les propriétés d'une tâche de détection de langue dominante asynchrone.

```
aws comprehend describe-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "DominantLanguageDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "languageanalysis1",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDominantLanguageDetectionJobà la section Référence des AWS CLI commandes.

`describe-endpoint`

L'exemple de code suivant montre comment utiliser`describe-endpoint`.

AWS CLI

Pour décrire un point de terminaison spécifique

L'`describe-endpoint`exemple suivant obtient les propriétés d'un point de terminaison spécifique au modèle.

```
aws comprehend describe-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint
```

Sortie :

```
{
    "EndpointProperties": {
        "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint,
        "Status": "IN_SERVICE",
        "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier1",
        "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier1",
        "DesiredInferenceUnits": 1,
        "CurrentInferenceUnits": 1,
        "CreationTime": "2023-06-13T20:32:54.526000+00:00",
        "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEndpointà la section Référence des AWS CLI commandes.

`describe-entities-detection-job`

L'exemple de code suivant montre comment utiliser`describe-entities-detection-job`.

AWS CLI

Pour décrire une tâche de détection d'entités

L'`describe-entities-detection-job`exemple suivant obtient les propriétés d'une tâche de détection d'entités asynchrones.

```
aws comprehend describe-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "EntitiesDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example-entity-detector",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
        "EndTime": "2023-06-08T21:40:23.509000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEntitiesDetectionJobà la section Référence des AWS CLI commandes.

`describe-entity-recognizer`

L'exemple de code suivant montre comment utiliser`describe-entity-recognizer`.

AWS CLI

Pour décrire un outil de reconnaissance d'entités

L'`describe-entity-recognizer`exemple suivant permet d'obtenir les propriétés d'un modèle de reconnaissance d'entités personnalisé.

```
aws comprehend describe-entity-recognizer \
    entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/business-recongizer-1/version/1
```

Sortie :

```
{
    "EntityRecognizerProperties": {
        "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/business-recongizer-1/version/1",
        "LanguageCode": "en",
        "Status": "TRAINED",
        "SubmitTime": "2023-06-14T20:44:59.631000+00:00",
        "EndTime": "2023-06-14T20:59:19.532000+00:00",
        "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",
        "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",
        "InputDataConfig": {
            "DataFormat": "COMPREHEND_CSV",
            "EntityTypes": [
                {
                    "Type": "BUSINESS"
                }
            ],
            "Documents": {
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/dataset/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "EntityList": {
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/entity.csv"
            }
        },
        "RecognizerMetadata": {
```

```
            "NumberOfTrainedDocuments": 1814,
            "NumberOfTestDocuments": 486,
            "EvaluationMetrics": {
                "Precision": 100.0,
                "Recall": 100.0,
                "F1Score": 100.0
            },
            "EntityTypes": [
                {
                    "Type": "BUSINESS",
                    "EvaluationMetrics": {
                        "Precision": 100.0,
                        "Recall": 100.0,
                        "F1Score": 100.0
                    },
                    "NumberOfTrainMentions": 1520
                }
            ]
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "VersionName": "1"
    }
}
```

Pour plus d'informations, consultez la section Reconnaissance d'entités personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEntityRecognizerà la section Référence des AWS CLI commandes.

`describe-events-detection-job`

L'exemple de code suivant montre comment utiliser`describe-events-detection-job`.

AWS CLI

Pour décrire une tâche de détection d'événements.

L'`describe-events-detection-job`exemple suivant permet d'obtenir les propriétés d'une tâche de détection d'événements asynchrones.

```
aws comprehend describe-events-detection-job \
```

```
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "EventsDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "events_job_1",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-12T18:45:56.054000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/EventsData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
EVENTS-123456abcdeb0e11022f22a11EXAMPLE/output/"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "TargetEventTypes": [
            "BANKRUPTCY",
            "EMPLOYMENT",
            "CORPORATE_ACQUISITION",
            "CORPORATE_MERGER",
            "INVESTMENT_GENERAL"
        ]
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEventsDetectionJobà la section Référence des AWS CLI commandes.

`describe-flywheel-iteration`

L'exemple de code suivant montre comment utiliser`describe-flywheel-iteration`.

AWS CLI

Pour décrire une itération en volant

L'`describe-flywheel-iteration`exemple suivant obtient les propriétés d'une itération au volant.

```
aws comprehend describe-flywheel-iteration \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel \
    --flywheel-iteration-id 20232222AEXAMPLE
```

Sortie :

```
{
    "FlywheelIterationProperties": {
        "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity",
        "FlywheelIterationId": "20232222AEXAMPLE",
        "CreationTime": "2023-06-16T21:10:26.385000+00:00",
        "EndTime": "2023-06-16T23:33:16.827000+00:00",
        "Status": "COMPLETED",
        "Message": "FULL_ITERATION: Flywheel iteration performed all functions
 successfully.",
        "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/1",
        "EvaluatedModelMetrics": {
            "AverageF1Score": 0.7742663922375772,
            "AveragePrecision": 0.8287636394041166,
            "AverageRecall": 0.7427084833645399,
            "AverageAccuracy": 0.8795394154118689
        },
        "TrainedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/Comprehend-Generated-v1-bb52d585",
        "TrainedModelMetrics": {
            "AverageF1Score": 0.9767700253081214,
            "AveragePrecision": 0.9767700253081214,
            "AverageRecall": 0.9767700253081214,
            "AverageAccuracy": 0.9858281665190434
        },
        "EvaluationManifestS3Prefix": "s3://amzn-s3-demo-destination-bucket/
flywheel-entity/schemaVersion=1/20230616T200543Z/evaluation/20230616T211026Z/"
    }
```

```
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeFlywheelIterationà la section Référence des AWS CLI commandes.

`describe-flywheel`

L'exemple de code suivant montre comment utiliser`describe-flywheel`.

AWS CLI

Pour décrire un volant

L'`describe-flywheel`exemple suivant permet d'obtenir les propriétés d'un volant d'inertie. Dans cet exemple, le modèle associé au volant est un modèle de classificateur personnalisé conçu pour classer les documents en tant que spam ou non-spam, ou en tant que « jambon ».

```
aws comprehend describe-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel
```

Sortie :

```
{
    "FlywheelProperties": {
        "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel",
        "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-model/version/1",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "TaskConfig": {
            "LanguageCode": "en",
            "DocumentClassificationConfig": {
                "Mode": "MULTI_CLASS",
                "Labels": [
                    "ham",
                    "spam"
                ]
```

```
            }
        },
        "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/example-flywheel/
schemaVersion=1/20230616T200543Z/",
        "DataSecurityConfig": {},
        "Status": "ACTIVE",
        "ModelType": "DOCUMENT_CLASSIFIER",
        "CreationTime": "2023-06-16T20:05:43.242000+00:00",
        "LastModifiedTime": "2023-06-16T20:21:43.567000+00:00"
    }
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le guide du développeur Amazon Comprehend.

- Pour plus de détails sur l'API, reportez-vous DescribeFlywheelà la section Référence des AWS CLI commandes.

`describe-key-phrases-detection-job`

L'exemple de code suivant montre comment utiliser`describe-key-phrases-detection-job`.

AWS CLI

Pour décrire une tâche de détection de phrases clés

L'`describe-key-phrases-detection-job`exemple suivant obtient les propriétés d'une tâche de détection de phrases-clés asynchrones.

```
aws comprehend describe-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "KeyPhrasesDetectionJobProperties": {
        "JobId": "69aa080c00fc68934a6a98f10EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/69aa080c00fc68934a6a98f10EXAMPLE",
        "JobName": "example-key-phrases-detection-job",
        "JobStatus": "COMPLETED",
        "SubmitTime": 1686606439.177,
        "EndTime": 1686606806.157,
```

```
        "InputDataConfig": {
            "S3Uri": "s3://dereksbucket1001/EventsData/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://dereksbucket1002/testfolder/111122223333-
KP-69aa080c00fc68934a6a98f10EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testrole"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeKeyPhrasesDetectionJobà la section Référence des AWS CLI commandes.

`describe-pii-entities-detection-job`

L'exemple de code suivant montre comment utiliser`describe-pii-entities-detection-job`.

AWS CLI

Pour décrire une tâche de détection d'entités PII

L'`describe-pii-entities-detection-job`exemple suivant obtient les propriétés d'une tâche de détection d'entités pii asynchrones.

```
aws comprehend describe-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "PiiEntitiesDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example-pii-entities-job",
        "JobStatus": "IN_PROGRESS",
```

```
        "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
        "EndTime": "2023-06-08T21:40:23.509000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribePiiEntitiesDetectionJobà la section Référence des AWS CLI commandes.

`describe-resource-policy`

L'exemple de code suivant montre comment utiliser`describe-resource-policy`.

AWS CLI

Pour décrire une politique de ressources associée à un modèle

L'`describe-resource-policy`exemple suivant obtient les propriétés d'une politique basée sur les ressources attachée à un modèle.

```
aws comprehend describe-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

Sortie :

```
{
    "ResourcePolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":
\"comprehend:ImportModel\",\"Resource\":\"*\"}]}",
```

```
    "CreationTime": "2023-06-19T18:44:26.028000+00:00",
    "LastModifiedTime": "2023-06-19T18:53:02.002000+00:00",
    "PolicyRevisionId": "baa675d069d07afaa2aa3106ae280f61"
}
```

Pour plus d'informations, consultez Copier des modèles personnalisés entre AWS comptes dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeResourcePolicyà la section Référence des AWS CLI commandes.

`describe-sentiment-detection-job`

L'exemple de code suivant montre comment utiliser`describe-sentiment-detection-job`.

AWS CLI

Pour décrire une tâche de détection des sentiments

L'`describe-sentiment-detection-job`exemple suivant permet d'obtenir les propriétés d'une tâche de détection de sentiments asynchrone.

```
aws comprehend describe-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "SentimentDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "movie_review_analysis",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
```

```
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeSentimentDetectionJobà la section Référence des AWS CLI commandes.

`describe-targeted-sentiment-detection-job`

L'exemple de code suivant montre comment utiliser`describe-targeted-sentiment-detection-job`.

AWS CLI

Pour décrire une tâche de détection ciblée des sentiments

L'`describe-targeted-sentiment-detection-job`exemple suivant permet d'obtenir les propriétés d'une tâche de détection ciblée asynchrone des sentiments.

```
aws comprehend describe-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "TargetedSentimentDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "movie_review_analysis",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
```

```
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeTargetedSentimentDetectionJobà la section Référence des AWS CLI commandes.

`describe-topics-detection-job`

L'exemple de code suivant montre comment utiliser`describe-topics-detection-job`.

AWS CLI

Pour décrire une tâche de détection de sujets

L'`describe-topics-detection-job`exemple suivant obtient les propriétés d'une tâche de détection de sujets asynchrones.

```
aws comprehend describe-topics-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "TopicsDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example_topics_detection",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket",
```

```
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "NumberOfTopics": 10,
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-examplerole"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeTopicsDetectionJobà la section Référence des AWS CLI commandes.

`detect-dominant-language`

L'exemple de code suivant montre comment utiliser`detect-dominant-language`.

AWS CLI

Pour détecter la langue dominante du texte saisi

Ce qui suit `detect-dominant-language` analyse le texte saisi et identifie la langue dominante. Le score de confiance du modèle préentraîné est également affiché.

```
aws comprehend detect-dominant-language \
    --text "It is a beautiful day in Seattle."
```

Sortie :

```
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9877256155014038
        }
    ]
}
```

Pour plus d'informations, consultez Dominant Language dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectDominantLanguageà la section Référence des AWS CLI commandes.

`detect-entities`

L'exemple de code suivant montre comment utiliser`detect-entities`.

AWS CLI

Pour détecter les entités nommées dans le texte saisi

L'`detect-entities`exemple suivant analyse le texte saisi et renvoie les entités nommées. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
    account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st. Based on your autopay settings, \
    we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000. \
    Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

Sortie :

```
{
    "Entities": [
        {
            "Score": 0.9994556307792664,
            "Type": "PERSON",
            "Text": "Zhang Wei",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9981022477149963,
            "Type": "PERSON",
            "Text": "John",
```

```
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.9986887574195862,
            "Type": "ORGANIZATION",
            "Text": "AnyCompany Financial Services, LLC",
            "BeginOffset": 33,
            "EndOffset": 67
        },
        {
            "Score": 0.9959119558334351,
            "Type": "OTHER",
            "Text": "1111-XXXX-1111-XXXX",
            "BeginOffset": 88,
            "EndOffset": 107
        },
        {
            "Score": 0.9708039164543152,
            "Type": "QUANTITY",
            "Text": ".53",
            "BeginOffset": 133,
            "EndOffset": 136
        },
        {
            "Score": 0.9987268447875977,
            "Type": "DATE",
            "Text": "July 31st",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9858865737915039,
            "Type": "OTHER",
            "Text": "XXXXXX1111",
            "BeginOffset": 271,
            "EndOffset": 281
        },
        {
            "Score": 0.9700471758842468,
            "Type": "OTHER",
            "Text": "XXXXX0000",
            "BeginOffset": 306,
            "EndOffset": 315
```

```
        },
        {
            "Score": 0.9591118693351746,
            "Type": "ORGANIZATION",
            "Text": "Sunshine Spa",
            "BeginOffset": 340,
            "EndOffset": 352
        },
        {
            "Score": 0.9797496795654297,
            "Type": "LOCATION",
            "Text": "123 Main St",
            "BeginOffset": 354,
            "EndOffset": 365
        },
        {
            "Score": 0.994929313659668,
            "Type": "PERSON",
            "Text": "Alice",
            "BeginOffset": 394,
            "EndOffset": 399
        },
        {
            "Score": 0.9949769377708435,
            "Type": "OTHER",
            "Text": "AnySpa@example.com",
            "BeginOffset": 403,
            "EndOffset": 418
        }
    ]
}
```

Pour plus d'informations, consultez Entities dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectEntitiesà la section Référence des AWS CLI commandes.

`detect-key-phrases`

L'exemple de code suivant montre comment utiliser`detect-key-phrases`.

AWS CLI

Pour détecter les phrases clés dans le texte saisi

L'`detect-key-phrases`exemple suivant analyse le texte saisi et identifie les phrases nominales clés. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-key-phrases \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card \
        account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July 31st. Based on your autopay settings, \
        we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000. \
        Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "KeyPhrases": [
        {
            "Score": 0.8996376395225525,
            "Text": "Zhang Wei",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9992469549179077,
            "Text": "John",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.988385021686554,
            "Text": "Your AnyCompany Financial Services",
            "BeginOffset": 28,
            "EndOffset": 62
        },
        {
            "Score": 0.8740853071212769,
            "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
```

```
            "BeginOffset": 64,
            "EndOffset": 107
        },
        {
            "Score": 0.9999437928199768,
            "Text": "a minimum payment",
            "BeginOffset": 112,
            "EndOffset": 129
        },
        {
            "Score": 0.9998900890350342,
            "Text": ".53",
            "BeginOffset": 133,
            "EndOffset": 136
        },
        {
            "Score": 0.9979453086853027,
            "Text": "July 31st",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9983011484146118,
            "Text": "your autopay settings",
            "BeginOffset": 172,
            "EndOffset": 193
        },
        {
            "Score": 0.9996572136878967,
            "Text": "your payment",
            "BeginOffset": 211,
            "EndOffset": 223
        },
        {
            "Score": 0.9995037317276001,
            "Text": "the due date",
            "BeginOffset": 227,
            "EndOffset": 239
        },
        {
            "Score": 0.9702621698379517,
            "Text": "your bank account number XXXXXX1111",
            "BeginOffset": 245,
            "EndOffset": 280
```

```
        },
        {
            "Score": 0.9179925918579102,
            "Text": "the routing number XXXXX0000.Customer feedback",
            "BeginOffset": 286,
            "EndOffset": 332
        },
        {
            "Score": 0.9978160858154297,
            "Text": "Sunshine Spa",
            "BeginOffset": 337,
            "EndOffset": 349
        },
        {
            "Score": 0.9706913232803345,
            "Text": "123 Main St",
            "BeginOffset": 351,
            "EndOffset": 362
        },
        {
            "Score": 0.9941995143890381,
            "Text": "comments",
            "BeginOffset": 379,
            "EndOffset": 387
        },
        {
            "Score": 0.9759287238121033,
            "Text": "Alice",
            "BeginOffset": 391,
            "EndOffset": 396
        },
        {
            "Score": 0.8376792669296265,
            "Text": "AnySpa@example.com",
            "BeginOffset": 400,
            "EndOffset": 415
        }
    ]
}
```


Pour plus d'informations, consultez les phrases clés du manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectKeyPhrasesà la section Référence des AWS CLI commandes.

`detect-pii-entities`

L'exemple de code suivant montre comment utiliser`detect-pii-entities`.

AWS CLI

Pour détecter les entités pii dans le texte saisi

L'`detect-pii-entities`exemple suivant analyse le texte saisi et identifie les entités contenant des informations personnelles identifiables (PII). Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-pii-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
       account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
 July 31st. Based on your autopay settings, \
       we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000. \
       Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

Sortie :

```
{
    "Entities": [
        {
            "Score": 0.9998322129249573,
            "Type": "NAME",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9998878240585327,
            "Type": "NAME",
            "BeginOffset": 22,
            "EndOffset": 26
        },
```

```
        {
            "Score": 0.9994089603424072,
            "Type": "CREDIT_DEBIT_NUMBER",
            "BeginOffset": 88,
            "EndOffset": 107
        },
        {
            "Score": 0.9999760985374451,
            "Type": "DATE_TIME",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9999449253082275,
            "Type": "BANK_ACCOUNT_NUMBER",
            "BeginOffset": 271,
            "EndOffset": 281
        },
        {
            "Score": 0.9999847412109375,
            "Type": "BANK_ROUTING",
            "BeginOffset": 306,
            "EndOffset": 315
        },
        {
            "Score": 0.999925434589386,
            "Type": "ADDRESS",
            "BeginOffset": 354,
            "EndOffset": 365
        },
        {
            "Score": 0.9989161491394043,
            "Type": "NAME",
            "BeginOffset": 394,
            "EndOffset": 399
        },
        {
            "Score": 0.9994171857833862,
            "Type": "EMAIL",
            "BeginOffset": 403,
            "EndOffset": 418
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Informations personnelles identifiables (PII) dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectPiiEntitiesà la section Référence des AWS CLI commandes.

`detect-sentiment`

L'exemple de code suivant montre comment utiliser`detect-sentiment`.

AWS CLI

Pour détecter le sentiment d'un texte saisi

L'`detect-sentiment`exemple suivant analyse le texte saisi et renvoie une inférence du sentiment dominant (`POSITIVE`, `NEUTRALMIXED`, ou`NEGATIVE`).

```
aws comprehend detect-sentiment \
    --language-code en \
    --text "It is a beautiful day in Seattle"
```

Sortie :

```
{
    "Sentiment": "POSITIVE",
    "SentimentScore": {
        "Positive": 0.9976957440376282,
        "Negative": 9.653854067437351e-05,
        "Neutral": 0.002169104292988777,
        "Mixed": 3.857641786453314e-05
    }
}
```

Pour plus d'informations, consultez Sentiment dans le guide du développeur Amazon Comprehend

- Pour plus de détails sur l'API, reportez-vous DetectSentimentà la section Référence des AWS CLI commandes.

`detect-syntax`

L'exemple de code suivant montre comment utiliser`detect-syntax`.

AWS CLI

Pour détecter les parties du discours dans un texte saisi

L'`detect-syntax`exemple suivant analyse la syntaxe du texte saisi et renvoie les différentes parties du discours. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-syntax \
    --language-code en \
    --text "It is a beautiful day in Seattle."
```

Sortie :

```
{
    "SyntaxTokens": [
        {
            "TokenId": 1,
            "Text": "It",
            "BeginOffset": 0,
            "EndOffset": 2,
            "PartOfSpeech": {
                "Tag": "PRON",
                "Score": 0.9999740719795227
            }
        },
        {
            "TokenId": 2,
            "Text": "is",
            "BeginOffset": 3,
            "EndOffset": 5,
            "PartOfSpeech": {
                "Tag": "VERB",
                "Score": 0.999901294708252
            }
        },
        {
            "TokenId": 3,
            "Text": "a",
```

```
            "BeginOffset": 6,
            "EndOffset": 7,
            "PartOfSpeech": {
                "Tag": "DET",
                "Score": 0.9999938607215881
            }
        },
        {
            "TokenId": 4,
            "Text": "beautiful",
            "BeginOffset": 8,
            "EndOffset": 17,
            "PartOfSpeech": {
                "Tag": "ADJ",
                "Score": 0.9987351894378662
            }
        },
        {
            "TokenId": 5,
            "Text": "day",
            "BeginOffset": 18,
            "EndOffset": 21,
            "PartOfSpeech": {
                "Tag": "NOUN",
                "Score": 0.9999796748161316
            }
        },
        {
            "TokenId": 6,
            "Text": "in",
            "BeginOffset": 22,
            "EndOffset": 24,
            "PartOfSpeech": {
                "Tag": "ADP",
                "Score": 0.9998047947883606
            }
        },
        {
            "TokenId": 7,
            "Text": "Seattle",
            "BeginOffset": 25,
            "EndOffset": 32,
            "PartOfSpeech": {
                "Tag": "PROPN",
```

```
                "Score": 0.9940530061721802
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse syntaxique dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectSyntaxà la section Référence des AWS CLI commandes.

`detect-targeted-sentiment`

L'exemple de code suivant montre comment utiliser`detect-targeted-sentiment`.

AWS CLI

Pour détecter le sentiment ciblé des entités nommées dans un texte saisi

L'`detect-targeted-sentiment`exemple suivant analyse le texte saisi et renvoie les entités nommées en plus du sentiment ciblé associé à chaque entité. Le score de confiance des modèles préentraînés pour chaque prédiction est également généré.

```
aws comprehend detect-targeted-sentiment \
    --language-code en \
    --text "I do not enjoy January because it is too cold but August is the perfect
 temperature"
```

Sortie :

```
{
    "Entities": [
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.9999979734420776,
                    "GroupScore": 1.0,
                    "Text": "I",
```

```
                    "Type": "PERSON",
                    "MentionSentiment": {
                        "Sentiment": "NEUTRAL",
                        "SentimentScore": {
                            "Positive": 0.0,
                            "Negative": 0.0,
                            "Neutral": 1.0,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 0,
                    "EndOffset": 1
                }
            ]
        },
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.9638869762420654,
                    "GroupScore": 1.0,
                    "Text": "January",
                    "Type": "DATE",
                    "MentionSentiment": {
                        "Sentiment": "NEGATIVE",
                        "SentimentScore": {
                            "Positive": 0.0031610000878572464,
                            "Negative": 0.9967250227928162,
                            "Neutral": 0.00011100000119768083,
                            "Mixed": 1.9999999949504854e-06
                        }
                    },
                    "BeginOffset": 15,
                    "EndOffset": 22
                }
            ]
        },
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
```

```
                {
                {
                    "Score": 0.9664419889450073,
                    "GroupScore": 1.0,
                    "Text": "August",
                    "Type": "DATE",
                    "MentionSentiment": {
                        "Sentiment": "POSITIVE",
                        "SentimentScore": {
                            "Positive": 0.9999549984931946,
                            "Negative": 3.999999989900971e-06,
                            "Neutral": 4.099999932805076e-05,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 50,
                    "EndOffset": 56
                }
            ]
        },
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.9803199768066406,
                    "GroupScore": 1.0,
                    "Text": "temperature",
                    "Type": "ATTRIBUTE",
                    "MentionSentiment": {
                        "Sentiment": "POSITIVE",
                        "SentimentScore": {
                            "Positive": 1.0,
                            "Negative": 0.0,
                            "Neutral": 0.0,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 77,
                    "EndOffset": 88
                }
            ]
        }
```

```
    ]
}
```

Pour plus d'informations, consultez Targeted Sentiment dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectTargetedSentimentà la section Référence des AWS CLI commandes.

`import-model`

L'exemple de code suivant montre comment utiliser`import-model`.

AWS CLI

Pour importer un modèle

L'`import-model`exemple suivant importe un modèle depuis un autre AWS compte. Le modèle de classificateur de documents en compte `444455556666` dispose d'une politique basée sur les ressources permettant `111122223333` au compte d'importer le modèle.

```
aws comprehend import-model \
    --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-classifier/example-classifier
```

Sortie :

```
{
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier"
}
```

Pour plus d'informations, consultez Copier des modèles personnalisés entre AWS comptes dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ImportModelà la section Référence des AWS CLI commandes.

`list-datasets`

L'exemple de code suivant montre comment utiliser`list-datasets`.

AWS CLI

Pour répertorier tous les ensembles de données sur les volants

L'`list-datasets`exemple suivant répertorie tous les ensembles de données associés à un volant.

```
aws comprehend list-datasets \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity
```

Sortie :

```
{
    "DatasetPropertiesList": [
        {
            "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity/dataset/example-dataset-1",
            "DatasetName": "example-dataset-1",
            "DatasetType": "TRAIN",
            "DatasetS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/schemaVersion=1/20230616T200543Z/datasets/example-dataset-1/20230616T203710Z/",
            "Status": "CREATING",
            "CreationTime": "2023-06-16T20:37:10.400000+00:00"
        },
        {
            "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity/dataset/example-dataset-2",
            "DatasetName": "example-dataset-2",
            "DatasetType": "TRAIN",
            "DatasetS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/schemaVersion=1/20230616T200543Z/datasets/example-dataset-2/20230616T200607Z/",
            "Description": "TRAIN Dataset created by Flywheel creation.",
            "Status": "COMPLETED",
            "NumberOfDocuments": 5572,
            "CreationTime": "2023-06-16T20:06:07.722000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le guide du développeur Amazon Comprehend.

- Pour plus de détails sur l'API, reportez-vous ListDatasetsà la section Référence des AWS CLI commandes.

`list-document-classification-jobs`

L'exemple de code suivant montre comment utiliser`list-document-classification-jobs`.

AWS CLI

Pour répertorier tous les travaux de classification de documents

L'`list-document-classification-jobs`exemple suivant répertorie toutes les tâches de classification de documents.

```
aws comprehend list-document-classification-jobs
```

Sortie :

```
{
    "DocumentClassificationJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "exampleclassificationjob",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
            "EndTime": "2023-06-14T17:15:58.582000+00:00",
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/1234567890101-CLN-e758dd56b824aa717ceab551f11749fb/output/output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
```

```
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "exampleclassificationjob2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-14T17:22:39.829000+00:00",
            "EndTime": "2023-06-14T17:28:46.107000+00:00",
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/1234567890101-CLN-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDocumentClassificationJobsà la section Référence des AWS CLI commandes.

`list-document-classifier-summaries`

L'exemple de code suivant montre comment utiliser`list-document-classifier-summaries`.

AWS CLI

Pour répertorier les résumés de tous les classificateurs de documents créés

L'`list-document-classifier-summaries`exemple suivant répertorie tous les résumés des classificateurs de documents créés.

```
aws comprehend list-document-classifier-summaries
```

Sortie :

```
{
    "DocumentClassifierSummariesList": [
        {
            "DocumentClassifierName": "example-classifier-1",
            "NumberOfVersions": 1,
            "LatestVersionCreatedAt": "2023-06-13T22:07:59.825000+00:00",
            "LatestVersionName": "1",
            "LatestVersionStatus": "TRAINED"
        },
        {
            "DocumentClassifierName": "example-classifier-2",
            "NumberOfVersions": 2,
            "LatestVersionCreatedAt": "2023-06-13T21:54:59.589000+00:00",
            "LatestVersionName": "2",
            "LatestVersionStatus": "TRAINED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDocumentClassifierSummariesà la section Référence des AWS CLI commandes.

`list-document-classifiers`

L'exemple de code suivant montre comment utiliser`list-document-classifiers`.

AWS CLI

Pour afficher la liste de tous les classificateurs de documents

L'`list-document-classifiers`exemple suivant répertorie tous les modèles de classificateur de documents entraînés et en cours de formation.

```
aws comprehend list-document-classifiers
```

Sortie :

```
{
    "DocumentClassifierPropertiesList": [
```

```
        {
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier1",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
            "EndTime": "2023-06-13T19:42:31.752000+00:00",
            "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
            "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
            "ClassifierMetadata": {
                "NumberOfLabels": 3,
                "NumberOfTrainedDocuments": 5016,
                "NumberOfTestDocuments": 557,
                "EvaluationMetrics": {
                    "Accuracy": 0.9856,
                    "Precision": 0.9919,
                    "Recall": 0.9459,
                    "F1Score": 0.9673,
                    "MicroPrecision": 0.9856,
                    "MicroRecall": 0.9856,
                    "MicroF1Score": 0.9856,
                    "HammingLoss": 0.0144
                }
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        },
        {
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier2",
            "LanguageCode": "en",
            "Status": "TRAINING",
            "SubmitTime": "2023-06-13T21:20:28.690000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
```

```
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDocumentClassifiersà la section Référence des AWS CLI commandes.

`list-dominant-language-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-dominant-language-detection-jobs`.

AWS CLI

Pour répertorier tous les emplois de détection des langues dominantes

L'`list-dominant-language-detection-jobs`exemple suivant répertorie toutes les tâches de détection du langage dominant asynchrone en cours et terminées.

```
aws comprehend list-dominant-language-detection-jobs
```

Sortie :

```
{
    "DominantLanguageDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "languageanalysis1",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
            "EndTime": "2023-06-09T18:18:45.498000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
```

```
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "languageanalysis2",
            "JobStatus": "STOPPED",
            "SubmitTime": "2023-06-09T18:16:33.690000+00:00",
            "EndTime": "2023-06-09T18:24:40.608000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDominantLanguageDetectionJobsà la section Référence des AWS CLI commandes.

`list-endpoints`

L'exemple de code suivant montre comment utiliser`list-endpoints`.

AWS CLI

Vers la liste de tous les points de terminaison

L'`list-endpoints`exemple suivant répertorie tous les points de terminaison spécifiques au modèle actif.

```
aws comprehend list-endpoints
```

Sortie :

```
{
    "EndpointPropertiesList": [
        {
            "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/ExampleClassifierEndpoint",
            "Status": "IN_SERVICE",
            "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
            "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
            "DesiredInferenceUnits": 1,
            "CurrentInferenceUnits": 1,
            "CreationTime": "2023-06-13T20:32:54.526000+00:00",
            "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
        },
        {
            "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/ExampleClassifierEndpoint2",
            "Status": "IN_SERVICE",
            "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2",
            "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2",
            "DesiredInferenceUnits": 1,
            "CurrentInferenceUnits": 1,
            "CreationTime": "2023-06-13T20:32:54.526000+00:00",
            "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListEndpointsà la section Référence des AWS CLI commandes.

`list-entities-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-entities-detection-jobs`.

AWS CLI

Pour répertorier toutes les tâches de détection d'entités

L'`list-entities-detection-jobs`exemple suivant répertorie toutes les tâches de détection d'entités asynchrones.

```
aws comprehend list-entities-detection-jobs
```

Sortie :

```
{
    "EntitiesDetectionJobPropertiesList": [
        {
            "JobId": "468af39c28ab45b83eb0c4ab9EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/468af39c28ab45b83eb0c4ab9EXAMPLE",
            "JobName": "example-entities-detection",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-08T20:57:46.476000+00:00",
            "EndTime": "2023-06-08T21:05:53.718000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-NER-468af39c28ab45b83eb0c4ab9EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
```

```
        {
            "JobId": "809691caeaab0e71406f80a28EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/809691caeaab0e71406f80a28EXAMPLE",
            "JobName": "example-entities-detection-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
            "EndTime": "2023-06-08T21:40:23.509000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-NER-809691caeaab0e71406f80a28EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "e00597c36b448b91d70dea165EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/e00597c36b448b91d70dea165EXAMPLE",
            "JobName": "example-entities-detection-3",
            "JobStatus": "STOPPED",
            "SubmitTime": "2023-06-08T22:19:28.528000+00:00",
            "EndTime": "2023-06-08T22:27:33.991000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-NER-e00597c36b448b91d70dea165EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez Entities dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListEntitiesDetectionJobsà la section Référence des AWS CLI commandes.

`list-entity-recognizer-summaries`

L'exemple de code suivant montre comment utiliser`list-entity-recognizer-summaries`.

AWS CLI

Vers la liste des résumés de tous les outils de reconnaissance d'entités créés

L'`list-entity-recognizer-summaries`exemple suivant répertorie tous les résumés des outils de reconnaissance d'entités.

```
aws comprehend list-entity-recognizer-summaries
```

Sortie :

```
{
    "EntityRecognizerSummariesList": [
        {
            "RecognizerName": "entity-recognizer-3",
            "NumberOfVersions": 2,
            "LatestVersionCreatedAt": "2023-06-15T23:15:07.621000+00:00",
            "LatestVersionName": "2",
            "LatestVersionStatus": "STOP_REQUESTED"
        },
        {
            "RecognizerName": "entity-recognizer-2",
            "NumberOfVersions": 1,
            "LatestVersionCreatedAt": "2023-06-14T22:55:27.805000+00:00",
            "LatestVersionName": "2"
            "LatestVersionStatus": "TRAINED"
        },
        {
            "RecognizerName": "entity-recognizer-1",
            "NumberOfVersions": 1,
            "LatestVersionCreatedAt": "2023-06-14T20:44:59.631000+00:00",
            "LatestVersionName": "1",
            "LatestVersionStatus": "TRAINED"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Reconnaissance d'entités personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListEntityRecognizerSummariesà la section Référence des AWS CLI commandes.

`list-entity-recognizers`

L'exemple de code suivant montre comment utiliser`list-entity-recognizers`.

AWS CLI

Pour répertorier tous les outils de reconnaissance d'entités personnalisés

L'`list-entity-recognizers`exemple suivant répertorie tous les outils de reconnaissance d'entités personnalisés créés.

```
aws comprehend list-entity-recognizers
```

Sortie :

```
{
    "EntityRecognizerPropertiesList": [
        {
            "EntityRecognizerArn": "arn:aws:comprehend:us-
west-2:111122223333:entity-recognizer/EntityRecognizer/version/1",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-14T20:44:59.631000+00:00",
            "EndTime": "2023-06-14T20:59:19.532000+00:00",
            "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",
            "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "EntityTypes": [
                    {
                        "Type": "BUSINESS"
                    }
                ],
```

```
                "Documents": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/dataset/",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "EntityList": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/entity.csv"
                }
            },
            "RecognizerMetadata": {
                "NumberOfTrainedDocuments": 1814,
                "NumberOfTestDocuments": 486,
                "EvaluationMetrics": {
                    "Precision": 100.0,
                    "Recall": 100.0,
                    "F1Score": 100.0
                },
                "EntityTypes": [
                    {
                        "Type": "BUSINESS",
                        "EvaluationMetrics": {
                            "Precision": 100.0,
                            "Recall": 100.0,
                            "F1Score": 100.0
                        },
                        "NumberOfTrainMentions": 1520
                    }
                ]
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole",
            "VersionName": "1"
        },
        {
            "EntityRecognizerArn": "arn:aws:comprehend:us-
west-2:111122223333:entity-recognizer/entityrecognizer3",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-14T22:57:51.056000+00:00",
            "EndTime": "2023-06-14T23:14:13.894000+00:00",
            "TrainingStartTime": "2023-06-14T23:01:33.984000+00:00",
            "TrainingEndTime": "2023-06-14T23:13:02.984000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "EntityTypes": [
```

```
                    {
                        "Type": "DEVICE"
                    }
                ],
                "Documents": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/raw_txt.csv",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "EntityList": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/entity_list.csv"
                }
            },
            "RecognizerMetadata": {
                "NumberOfTrainedDocuments": 4616,
                "NumberOfTestDocuments": 3489,
                "EvaluationMetrics": {
                    "Precision": 98.54227405247813,
                    "Recall": 100.0,
                    "F1Score": 99.26578560939794
                },
                "EntityTypes": [
                    {
                        "Type": "DEVICE",
                        "EvaluationMetrics": {
                            "Precision": 98.54227405247813,
                            "Recall": 100.0,
                            "F1Score": 99.26578560939794
                        },
                        "NumberOfTrainMentions": 2764
                    }
                ]
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
        }
    ]
}
```

Pour plus d'informations, consultez la section Reconnaissance d'entités personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListEntityRecognizersà la section Référence des AWS CLI commandes.

`list-events-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-events-detection-jobs`.

AWS CLI

Pour répertorier toutes les tâches de détection d'événements

L'`list-events-detection-jobs`exemple suivant répertorie toutes les tâches de détection d'événements asynchrones.

```
aws comprehend list-events-detection-jobs
```

Sortie :

```
{
    "EventsDetectionJobPropertiesList": [
        {
            "JobId": "aa9593f9203e84f3ef032ce18EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/aa9593f9203e84f3ef032ce18EXAMPLE",
            "JobName": "events_job_1",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-12T19:14:57.751000+00:00",
            "EndTime": "2023-06-12T19:21:04.962000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/EventsData/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/1111222233333-EVENTS-aa9593f9203e84f3ef032ce18EXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "TargetEventTypes": [
                "BANKRUPTCY",
                "EMPLOYMENT",
                "CORPORATE_ACQUISITION",
                "CORPORATE_MERGER",
                "INVESTMENT_GENERAL"
            ]
```

```
        },
        {
            "JobId": "4a990a2f7e82adfca6e171135EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/4a990a2f7e82adfca6e171135EXAMPLE",
            "JobName": "events_job_2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-12T19:55:43.702000+00:00",
            "EndTime": "2023-06-12T20:03:49.893000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/EventsData/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/1111222233333-EVENTS-4a990a2f7e82adfca6e171135EXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "TargetEventTypes": [
                "BANKRUPTCY",
                "EMPLOYMENT",
                "CORPORATE_ACQUISITION",
                "CORPORATE_MERGER",
                "INVESTMENT_GENERAL"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListEventsDetectionJobsà la section Référence des AWS CLI commandes.

`list-flywheel-iteration-history`

L'exemple de code suivant montre comment utiliser`list-flywheel-iteration-history`.

AWS CLI

Pour répertorier l'historique de toutes les itérations du volant

L'`list-flywheel-iteration-history`exemple suivant répertorie toutes les itérations d'un volant.

```
aws comprehend list-flywheel-iteration-history
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel
```

Sortie :

```
{
    "FlywheelIterationPropertiesList": [
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel",
            "FlywheelIterationId": "20230619TEXAMPLE",
            "CreationTime": "2023-06-19T04:00:32.594000+00:00",
            "EndTime": "2023-06-19T04:00:49.248000+00:00",
            "Status": "COMPLETED",
            "Message": "FULL_ITERATION: Flywheel iteration performed all functions
 successfully.",
            "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier/version/1",
            "EvaluatedModelMetrics": {
                "AverageF1Score": 0.7742663922375772,
                "AverageF1Score": 0.9876464664646313,
                "AveragePrecision": 0.9800000253081214,
                "AverageRecall": 0.9445600253081214,
                "AverageAccuracy": 0.9997281665190434
            },
            "EvaluationManifestS3Prefix": "s3://amzn-s3-demo-bucket/example-
flywheel/schemaVersion=1/20230619TEXAMPLE/evaluation/20230619TEXAMPLE/"
        },
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
            "FlywheelIterationId": "20230616TEXAMPLE",
            "CreationTime": "2023-06-16T21:10:26.385000+00:00",
            "EndTime": "2023-06-16T23:33:16.827000+00:00",
            "Status": "COMPLETED",
```

```
            "Message": "FULL_ITERATION: Flywheel iteration performed all functions
 successfully.",
            "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/spamvshamclassify/version/1",
            "EvaluatedModelMetrics": {
                "AverageF1Score": 0.7742663922375772,
                "AverageF1Score": 0.9767700253081214,
                "AveragePrecision": 0.9767700253081214,
                "AverageRecall": 0.9767700253081214,
                "AverageAccuracy": 0.9858281665190434
            },
            "EvaluationManifestS3Prefix": "s3://amzn-s3-demo-bucket/example-
flywheel-2/schemaVersion=1/20230616TEXAMPLE/evaluation/20230616TEXAMPLE/"
        }
    ]
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListFlywheelIterationHistoryà la section Référence des AWS CLI commandes.

`list-flywheels`

L'exemple de code suivant montre comment utiliser`list-flywheels`.

AWS CLI

Pour répertorier tous les volants

L'`list-flywheels`exemple suivant répertorie tous les volants créés.

```
aws comprehend list-flywheels
```

Sortie :

```
{
    "FlywheelSummaryList": [
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-1",
```

```
            "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier/version/1",
            "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/example-flywheel-1/
schemaVersion=1/20230616T200543Z/",
            "Status": "ACTIVE",
            "ModelType": "DOCUMENT_CLASSIFIER",
            "CreationTime": "2023-06-16T20:05:43.242000+00:00",
            "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
            "LatestFlywheelIteration": "20230619T040032Z"
        },
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
            "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2/version/1",
            "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/example-flywheel-2/
schemaVersion=1/20220616T200543Z/",
            "Status": "ACTIVE",
            "ModelType": "DOCUMENT_CLASSIFIER",
            "CreationTime": "2022-06-16T20:05:43.242000+00:00",
            "LastModifiedTime": "2022-06-19T04:00:43.027000+00:00",
            "LatestFlywheelIteration": "20220619T040032Z"
        }
    ]
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListFlywheelsà la section Référence des AWS CLI commandes.

`list-key-phrases-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-key-phrases-detection-jobs`.

AWS CLI

Pour répertorier toutes les tâches de détection de phrases clés

L'`list-key-phrases-detection-jobs`exemple suivant répertorie toutes les tâches de détection de phrases clés asynchrones en cours et terminées.

```
aws comprehend list-key-phrases-detection-jobs
```

Sortie :

```
{
    "KeyPhrasesDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "keyphrasesanalysis1",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-08T22:31:43.767000+00:00",
            "EndTime": "2023-06-08T22:39:52.565000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-KP-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a33EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a33EXAMPLE",
            "JobName": "keyphrasesanalysis2",
            "JobStatus": "STOPPED",
            "SubmitTime": "2023-06-08T22:57:52.154000+00:00",
            "EndTime": "2023-06-08T23:05:48.385000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-KP-123456abcdeb0e11022f22a33EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
```

```
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a44EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a44EXAMPLE",
            "JobName": "keyphrasesanalysis3",
            "JobStatus": "FAILED",
            "Message": "NO_READ_ACCESS_TO_INPUT: The provided data access role does
 not have proper access to the input data.",
            "SubmitTime": "2023-06-09T16:47:04.029000+00:00",
            "EndTime": "2023-06-09T16:47:18.413000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-KP-123456abcdeb0e11022f22a44EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListKeyPhrasesDetectionJobsà la section Référence des AWS CLI commandes.

`list-pii-entities-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-pii-entities-detection-jobs`.

AWS CLI

Pour répertorier toutes les tâches de détection d'entités pii

L'`list-pii-entities-detection-jobs`exemple suivant répertorie toutes les tâches de détection d'informations personnelles asynchrones en cours et terminées.

```
aws comprehend list-pii-entities-detection-jobs
```

Sortie :

```
{
    "PiiEntitiesDetectionJobPropertiesList": [
        {
            "JobId": "6f9db0c42d0c810e814670ee4EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/6f9db0c42d0c810e814670ee4EXAMPLE",
            "JobName": "example-pii-detection-job",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T21:02:46.241000+00:00",
            "EndTime": "2023-06-09T21:12:52.602000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/111122223333-
PII-6f9db0c42d0c810e814670ee4EXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "Mode": "ONLY_OFFSETS"
        },
        {
            "JobId": "d927562638cfa739331a99b3cEXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/d927562638cfa739331a99b3cEXAMPLE",
            "JobName": "example-pii-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T21:20:58.211000+00:00",
            "EndTime": "2023-06-09T21:31:06.027000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
```

```
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-PII-d927562638cfa739331a99b3cEXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "Mode": "ONLY_OFFSETS"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListPiiEntitiesDetectionJobsà la section Référence des AWS CLI commandes.

`list-sentiment-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-sentiment-detection-jobs`.

AWS CLI

Pour répertorier tous les emplois liés à la détection des sentiments

L'`list-sentiment-detection-jobs`exemple suivant répertorie toutes les tâches asynchrones de détection des sentiments en cours et terminées.

```
aws comprehend list-sentiment-detection-jobs
```

Sortie :

```
{
    "SentimentDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "example-sentiment-detection-job",
            "JobStatus": "IN_PROGRESS",
            "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
            "EndTime": "2023-06-09T22:52:27.416000+00:00",
            "InputDataConfig": {
```

```
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "example-sentiment-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
            "EndTime": "2023-06-09T23:26:00.168000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData2",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListSentimentDetectionJobsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises d'une ressource Amazon Comprehend.

```
aws comprehend list-tags-for-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

Sortie :

```
{
    "ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1",
    "Tags": [
        {
            "Key": "Department",
            "Value": "Finance"
        },
        {
            "Key": "location",
            "Value": "Seattle"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage de vos ressources dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-targeted-sentiment-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-targeted-sentiment-detection-jobs`.

AWS CLI

Pour répertorier toutes les tâches de détection de sentiments ciblées

L'`list-targeted-sentiment-detection-jobs`exemple suivant répertorie toutes les tâches de détection ciblée asynchrone des sentiments en cours et terminées.

```
aws comprehend list-targeted-sentiment-detection-jobs
```

Sortie :

```
{
    "TargetedSentimentDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "example-targeted-sentiment-detection-job",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
            "EndTime": "2023-06-09T22:52:27.416000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-IOrole"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "example-targeted-sentiment-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
            "EndTime": "2023-06-09T23:26:00.168000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData2",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
```

```
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTargetedSentimentDetectionJobsà la section Référence des AWS CLI commandes.

`list-topics-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-topics-detection-jobs`.

AWS CLI

Pour répertorier toutes les tâches de détection de sujets

L'`list-topics-detection-jobs`exemple suivant répertorie toutes les tâches de détection de sujets asynchrones en cours et terminées.

```
aws comprehend list-topics-detection-jobs
```

Sortie :

```
{
    "TopicsDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName" "topic-analysis-1"
            "JobStatus": "IN_PROGRESS",
            "SubmitTime": "2023-06-09T18:40:35.384000+00:00",
            "EndTime": "2023-06-09T18:46:41.936000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
```

```
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "topic-analysis-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
            "EndTime": "2023-06-09T18:50:50.872000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE3",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE3",
            "JobName": "topic-analysis-2",
            "JobStatus": "IN_PROGRESS",
            "SubmitTime": "2023-06-09T18:50:56.737000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE3/output/output.tar.gz"
```

```
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTopicsDetectionJobsà la section Référence des AWS CLI commandes.

`put-resource-policy`

L'exemple de code suivant montre comment utiliser`put-resource-policy`.

AWS CLI

Pour associer une politique basée sur les ressources

L'`put-resource-policy`exemple suivant associe une politique basée sur les ressources à un modèle afin qu'il puisse être importé par un autre AWS compte. La politique est attachée au modèle dans le compte `111122223333` et permet d'`444455556666`importer le modèle dans le compte.

```
aws comprehend put-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
    --resource-policy '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Action":"comprehend:ImportModel","Resource":"*","Principal":
{"AWS":["arn:aws:iam::444455556666:root"]}}]}'
```

Sortie :

```
{
    "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"
}
```

Pour plus d'informations, consultez Copier des modèles personnalisés entre AWS comptes dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutResourcePolicyà la section Référence des AWS CLI commandes.

`start-document-classification-job`

L'exemple de code suivant montre comment utiliser`start-document-classification-job`.

AWS CLI

Pour démarrer une tâche de classification de documents

L'`start-document-classification-job`exemple suivant démarre une tâche de classification de documents avec un modèle personnalisé sur tous les fichiers à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le compartiment S3 d'entrée contient `SampleSMStext1.txtSampleSMStext2.txt`, et`SampleSMStext3.txt`. Le modèle avait déjà été formé à la classification des documents contenant du spam et des messages non indésirables, ou « spam », des messages SMS. Lorsque le travail est terminé, `output.tar.gz` il est placé à l'emplacement spécifié par la `--output-data-config` balise. `output.tar.gzpredictions.jsonl`contient la liste de classification de chaque document. La sortie Json est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-document-classification-job \
    --job-name exampleclassificationjob \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket-INPUT/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/mymodel/version/12
```

Contenu de `SampleSMStext1.txt` :

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Contenu de `SampleSMStext2.txt` :

```
"Hi, when do you want me to pick you up from practice?"
```

Contenu de `SampleSMStext3.txt` :

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Sortie :

```
{
    "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu de `predictions.jsonl` :

```
{"File": "SampleSMSText1.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
 0.9999}, {"Name": "ham", "Score": 0.0001}]}
{"File": "SampleSMStext2.txt", "Line": "0", "Classes": [{"Name": "ham", "Score":
 0.9994}, {"Name": "spam", "Score": 0.0006}]}
{"File": "SampleSMSText3.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
 0.9999}, {"Name": "ham", "Score": 0.0001}]}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartDocumentClassificationJobà la section Référence des AWS CLI commandes.

`start-dominant-language-detection-job`

L'exemple de code suivant montre comment utiliser`start-dominant-language-detection-job`.

AWS CLI

Pour démarrer une tâche de détection de langue asynchrone

L'`start-dominant-language-detection-job`exemple suivant lance une tâche de détection de langue asynchrone pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le compartiment S3 contient`Sampletext1.txt`. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config`

balise. `output` Le dossier `output.txt` contient la langue dominante de chacun des fichiers texte ainsi que le score de confiance du modèle pré-entraîné pour chaque prédiction.

```
aws comprehend start-dominant-language-detection-job \
    --job-name example_language_analysis_job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Contenu du fichier Sampletext1.txt :

```
"Physics is the natural science that involves the study of matter and its motion and
 behavior through space and time, along with related concepts such as energy and
 force."
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu de `output.txt` :

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score":
 0.9913753867149353}], "Line": 0}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartDominantLanguageDetectionJobà la section Référence des AWS CLI commandes.

`start-entities-detection-job`

L'exemple de code suivant montre comment utiliser`start-entities-detection-job`.

AWS CLI

Exemple 1 : pour démarrer une tâche de détection d'entités standard à l'aide du modèle préentraîné

L'`start-entities-detection-job`exemple suivant lance une tâche de détection d'entités asynchrones pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le compartiment S3 contient `Sampletext1.txtSampletext2.txt`, et`Sampletext3.txt`. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config` balise. `output` Le dossier `output.txt` contient la liste de toutes les entités nommées détectées dans chaque fichier texte ainsi que le score de confiance du modèle pré-entraîné pour chaque prédiction. La sortie Json est imprimée sur une ligne par fichier d'entrée, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-entities-detection-job \
    --job-name entitiestest \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Contenu de `Sampletext1.txt` :

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st."
```

Contenu de `Sampletext2.txt` :

```
"Dear Max, based on your autopay settings for your account example1.org account, we
 will withdraw your payment on the due date from your bank account number XXXXXX1111
 with the routing number XXXXX0000. "
```

Contenu de `Sampletext3.txt` :

```
"Jane, please submit any customer feedback from this weekend to AnySpa, 123 Main St,
 Anywhere and send comments to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu `output.txt` avec des retraits de ligne pour plus de lisibilité :

```
{
"Entities": [
    {
    "BeginOffset": 6,
    "EndOffset": 15,
    "Score": 0.9994006636420306,
    "Text": "Zhang Wei",
    "Type": "PERSON"
    },
    {
    "BeginOffset": 22,
    "EndOffset": 26,
    "Score": 0.9976647915128143,
    "Text": "John",
    "Type": "PERSON"
    },
    {
    "BeginOffset": 33,
    "EndOffset": 67,
    "Score": 0.9984608700836206,
    "Text": "AnyCompany Financial Services, LLC",
    "Type": "ORGANIZATION"
    },
    {
    "BeginOffset": 88,
    "EndOffset": 107,
    "Score": 0.9868521019555556,
    "Text": "1111-XXXX-1111-XXXX",
    "Type": "OTHER"
    },
    {
    "BeginOffset": 133,
```

```
    "EndOffset": 139,
    "Score": 0.998242565709204,
    "Text": "$24.53",
    "Type": "QUANTITY"
    },
    {
    "BeginOffset": 155,
    "EndOffset": 164,
    "Score": 0.9993039263159287,
    "Text": "July 31st",
    "Type": "DATE"
    }
],
"File": "SampleText1.txt",
"Line": 0
}
{
"Entities": [
    {
    "BeginOffset": 5,
    "EndOffset": 8,
    "Score": 0.9866232147545232,
    "Text": "Max",
    "Type": "PERSON"
    },
    {
    "BeginOffset": 156,
    "EndOffset": 166,
    "Score": 0.9797723450933329,
    "Text": "XXXXXX1111",
    "Type": "OTHER"
    },
    {
    "BeginOffset": 191,
    "EndOffset": 200,
    "Score": 0.9247838572396843,
    "Text": "XXXXX0000",
    "Type": "OTHER"
    }
],
"File": "SampleText2.txt",
"Line": 0
}
{
```

```
  "Entities": [
     {
     "Score": 0.9990532994270325,
     "Type": "PERSON",
     "Text": "Jane",
     "BeginOffset": 0,
     "EndOffset": 4
     },
     {
     "Score": 0.9519651532173157,
     "Type": "DATE",
     "Text": "this weekend",
     "BeginOffset": 47,
     "EndOffset": 59
     },
     {
     "Score": 0.5566426515579224,
     "Type": "ORGANIZATION",
     "Text": "AnySpa",
     "BeginOffset": 63,
     "EndOffset": 69
     },
     {
     "Score": 0.8059805631637573,
     "Type": "LOCATION",
     "Text": "123 Main St, Anywhere",
     "BeginOffset": 71,
     "EndOffset": 92
     },
     {
     "Score": 0.998830258846283,
     "Type": "PERSON",
     "Text": "Alice",
     "BeginOffset": 114,
     "EndOffset": 119
     },
     {
     "Score": 0.997818112373352,
     "Type": "OTHER",
     "Text": "AnySpa@example.com",
     "BeginOffset": 123,
     "EndOffset": 138
     }
     ],
```

```
    "File": "SampleText3.txt",
    "Line": 0
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

Exemple 2 : pour démarrer une tâche de détection d'entités personnalisée

L'`start-entities-detection-job`exemple suivant lance une tâche de détection d'entités personnalisées asynchrones pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le compartiment S3 de cet exemple contient `SampleFeedback1.txtSampleFeedback2.txt`, et`SampleFeedback3.txt`. Le modèle de reconnaissance d'entités a été formé sur la base des commentaires du support client pour reconnaître les noms des appareils. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config` balise. `output` Le dossier contient `output.txt` la liste de toutes les entités nommées détectées dans chaque fichier texte ainsi que le score de confiance du modèle préentraîné pour chaque prédiction. La sortie Json est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-entities-detection-job \
    --job-name customentitiestest \
    --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/entityrecognizer" \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-IOrole"
```

Contenu de `SampleFeedback1.txt` :

```
"I've been on the AnyPhone app have had issues for 24 hours when trying to pay bill.
 Cannot make payment. Sigh. | Oh man! Lets get that app up and running. DM me, and
 we can get to work!"
```

Contenu de `SampleFeedback2.txt` :

```
"Hi, I have a discrepancy with my new bill. Could we get it sorted out? A rep added
 stuff I didnt sign up for when I did my AnyPhone 10 upgrade. | We can absolutely
 get this sorted!"
```

Contenu de `SampleFeedback3.txt` :

```
"Is the by 1 get 1 free AnySmartPhone promo still going on? | Hi Christian! It ended
 yesterday, send us a DM if you have any questions and we can take a look at your
 options!"
```

Sortie :

```
{
    "JobId": "019ea9edac758806850fa8a79ff83021",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/019ea9edac758806850fa8a79ff83021",
    "JobStatus": "SUBMITTED"
}
```

Contenu `output.txt` avec des retraits de ligne pour plus de lisibilité :

```
{
"Entities": [
    {
    "BeginOffset": 17,
    "EndOffset": 25,
    "Score": 0.9999728210205924,
    "Text": "AnyPhone",
    "Type": "DEVICE"
    }
],
"File": "SampleFeedback1.txt",
"Line": 0
}
{
"Entities": [
    {
    "BeginOffset": 123,
    "EndOffset": 133,
    "Score": 0.9999892116761524,
    "Text": "AnyPhone 10",
```

```
    "Type": "DEVICE"
    }
],
"File": "SampleFeedback2.txt",
"Line": 0
}
{
"Entities": [
    {
    "BeginOffset": 23,
    "EndOffset": 35,
    "Score": 0.9999971389852362,
    "Text": "AnySmartPhone",
    "Type": "DEVICE"
    }
],
"File": "SampleFeedback3.txt",
"Line": 0
}
```

Pour plus d'informations, consultez la section Reconnaissance d'entités personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartEntitiesDetectionJobà la section Référence des AWS CLI commandes.

`start-events-detection-job`

L'exemple de code suivant montre comment utiliser`start-events-detection-job`.

AWS CLI

Pour démarrer une tâche de détection d'événements asynchrones

L'`start-events-detection-job`exemple suivant lance une tâche de détection d'événements asynchrones pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Les types d'événements cibles possibles incluent `BANKRUPCTY EMPLOYMENTCORPORATE_ACQUISITION,,INVESTMENT_GENERAL,CORPORATE_MERGER,IPO,RIGHTS_` etSTOCK_SPLIT. Dans cet exemple, le compartiment S3 contient `SampleText1.txtSampleText2.txt`, et`SampleText3.txt`. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config` balise. `output` Le dossier contient `SampleText1.txt.outSampleText2.txt.out`, et`SampleText3.txt.out`.

La sortie JSON est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-events-detection-job \
    --job-name events-detection-1 \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/EventsData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole \
    --language-code en \
    --target-event-
types "BANKRUPTCY" "EMPLOYMENT" "CORPORATE_ACQUISITION" "CORPORATE_MERGER" "INVESTMENT_GENER
```

Contenu de `SampleText1.txt` :

```
"Company AnyCompany grew by increasing sales and through acquisitions. After
 purchasing competing firms in 2020, AnyBusiness, a part of the AnyBusinessGroup,
 gave Jane Does firm a going rate of one cent a gallon or forty-two cents a barrel."
```

Contenu de `SampleText2.txt` :

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars,
 surprising and exciting the shareholders."
```

Contenu de `SampleText3.txt` :

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for
 bankruptcy."
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu `SampleText1.txt.out` avec des retraits de ligne pour plus de lisibilité :

```
{
```

```
    "Entities": [
        {
        "Mentions": [
            {
            "BeginOffset": 8,
            "EndOffset": 18,
            "Score": 0.99977,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            },
            {
            "BeginOffset": 112,
            "EndOffset": 123,
            "Score": 0.999747,
            "Text": "AnyBusiness",
            "Type": "ORGANIZATION",
            "GroupScore": 0.979826
            },
            {
            "BeginOffset": 171,
            "EndOffset": 175,
            "Score": 0.999615,
            "Text": "firm",
            "Type": "ORGANIZATION",
            "GroupScore": 0.871647
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 97,
            "EndOffset": 102,
            "Score": 0.987687,
            "Text": "firms",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
```

```
            "BeginOffset": 103,
            "EndOffset": 110,
            "Score": 0.999458,
            "Text": "in 2020",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 160,
            "EndOffset": 168,
            "Score": 0.999649,
            "Text": "John Doe",
            "Type": "PERSON",
            "GroupScore": 1
            }
        ]
        }
    ],
    "Events": [
        {
        "Type": "CORPORATE_ACQUISITION",
        "Arguments": [
            {
            "EntityIndex": 0,
            "Role": "INVESTOR",
            "Score": 0.99977
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 56,
            "EndOffset": 68,
            "Score": 0.999967,
            "Text": "acquisitions",
            "Type": "CORPORATE_ACQUISITION",
            "GroupScore": 1
            }
        ]
        },
        {
```

```
        "Type": "CORPORATE_ACQUISITION",
        "Arguments": [
            {
            "EntityIndex": 1,
            "Role": "INVESTEE",
            "Score": 0.987687
            },
            {
            "EntityIndex": 2,
            "Role": "DATE",
            "Score": 0.999458
            },
            {
            "EntityIndex": 3,
            "Role": "INVESTOR",
            "Score": 0.999649
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 76,
            "EndOffset": 86,
            "Score": 0.999973,
            "Text": "purchasing",
            "Type": "CORPORATE_ACQUISITION",
            "GroupScore": 1
            }
        ]
        }
    ],
    "File": "SampleText1.txt",
    "Line": 0
}
```

Contenu de `SampleText2.txt.out` :

```
{
    "Entities": [
        {
        "Mentions": [
            {
            "BeginOffset": 0,
            "EndOffset": 7,
```

```
            "Score": 0.999473,
            "Text": "In 2021",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 9,
            "EndOffset": 19,
            "Score": 0.999636,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 45,
            "EndOffset": 56,
            "Score": 0.999712,
            "Text": "AnyBusiness",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 61,
            "EndOffset": 80,
            "Score": 0.998886,
            "Text": "100 billion dollars",
            "Type": "MONETARY_VALUE",
            "GroupScore": 1
            }
        ]
        }
    ],
```

```
    "Events": [
        {
        "Type": "CORPORATE_ACQUISITION",
        "Arguments": [
            {
            "EntityIndex": 3,
            "Role": "AMOUNT",
            "Score": 0.998886
            },
            {
            "EntityIndex": 2,
            "Role": "INVESTEE",
            "Score": 0.999712
            },
            {
            "EntityIndex": 0,
            "Role": "DATE",
            "Score": 0.999473
            },
            {
            "EntityIndex": 1,
            "Role": "INVESTOR",
            "Score": 0.999636
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 31,
            "EndOffset": 40,
            "Score": 0.99995,
            "Text": "purchased",
            "Type": "CORPORATE_ACQUISITION",
            "GroupScore": 1
            }
        ]
        }
    ],
    "File": "SampleText2.txt",
    "Line": 0
}
```


Contenu de `SampleText3.txt.out` :

```
{
    "Entities": [
        {
        "Mentions": [
            {
            "BeginOffset": 9,
            "EndOffset": 19,
            "Score": 0.999774,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            },
            {
            "BeginOffset": 66,
            "EndOffset": 70,
            "Score": 0.995717,
            "Text": "they",
            "Type": "ORGANIZATION",
            "GroupScore": 0.997626
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 50,
            "EndOffset": 65,
            "Score": 0.999656,
            "Text": "later that year",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        }
    ],
    "Events": [
        {
        "Type": "BANKRUPTCY",
        "Arguments": [
            {
            "EntityIndex": 1,
            "Role": "DATE",
            "Score": 0.999656
```

```
            },
            {
            "EntityIndex": 0,
            "Role": "FILER",
            "Score": 0.995717
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 81,
            "EndOffset": 91,
            "Score": 0.999936,
            "Text": "bankruptcy",
            "Type": "BANKRUPTCY",
            "GroupScore": 1
            }
        ]
        }
    ],
    "File": "SampleText3.txt",
    "Line": 0
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartEventsDetectionJobà la section Référence des AWS CLI commandes.

`start-flywheel-iteration`

L'exemple de code suivant montre comment utiliser`start-flywheel-iteration`.

AWS CLI

Pour démarrer une itération en volant

L'`start-flywheel-iteration`exemple suivant lance une itération au volant. Cette opération utilise tous les nouveaux ensembles de données présents dans le volant pour entraîner une nouvelle version du modèle.

```
aws comprehend start-flywheel-iteration \
```

```
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel
```

Sortie :

```
{
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel",
    "FlywheelIterationId": "12345123TEXAMPLE"
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartFlywheelIterationà la section Référence des AWS CLI commandes.

`start-key-phrases-detection-job`

L'exemple de code suivant montre comment utiliser`start-key-phrases-detection-job`.

AWS CLI

Pour démarrer une tâche de détection de phrases clés

L'`start-key-phrases-detection-job`exemple suivant lance une tâche de détection de phrases clés asynchrones pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le compartiment S3 contient `Sampletext1.txtSampletext2.txt`, et`Sampletext3.txt`. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config` balise. `output` Le dossier contient le fichier `output.txt` qui contient toutes les phrases clés détectées dans chaque fichier texte et le score de confiance du modèle préentraîné pour chaque prédiction. La sortie Json est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-key-phrases-detection-job \
    --job-name keyphrasesanalysistest1 \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
```

```
    --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role" \
    --language-code en
```

Contenu de `Sampletext1.txt` :

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st."
```

Contenu de `Sampletext2.txt` :

```
"Dear Max, based on your autopay settings for your account Internet.org account, we
 will withdraw your payment on the due date from your bank account number XXXXXX1111
 with the routing number XXXXX0000. "
```

Contenu de `Sampletext3.txt` :

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123
 Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu `output.txt` avec des retraits de ligne pour plus de lisibilité :

```
{
    "File": "SampleText1.txt",
    "KeyPhrases": [
        {
        "BeginOffset": 6,
        "EndOffset": 15,
        "Score": 0.9748965572679326,
        "Text": "Zhang Wei"
        },
```

```
        {
        "BeginOffset": 22,
        "EndOffset": 26,
        "Score": 0.9997344722354619,
        "Text": "John"
        },
        {
        "BeginOffset": 28,
        "EndOffset": 62,
        "Score": 0.9843791074032948,
        "Text": "Your AnyCompany Financial Services"
        },
        {
        "BeginOffset": 64,
        "EndOffset": 107,
        "Score": 0.8976122401721824,
        "Text": "LLC credit card account 1111-XXXX-1111-XXXX"
        },
        {
        "BeginOffset": 112,
        "EndOffset": 129,
        "Score": 0.9999612982629748,
        "Text": "a minimum payment"
        },
        {
        "BeginOffset": 133,
        "EndOffset": 139,
        "Score": 0.99975728947036,
        "Text": "$24.53"
        },
        {
        "BeginOffset": 155,
        "EndOffset": 164,
        "Score": 0.9940866241449973,
        "Text": "July 31st"
        }
    ],
    "Line": 0
    }
    {
    "File": "SampleText2.txt",
    "KeyPhrases": [
        {
        "BeginOffset": 0,
```

```
        "EndOffset": 8,
        "Score": 0.9974021100118472,
        "Text": "Dear Max"
        },
        {
        "BeginOffset": 19,
        "EndOffset": 40,
        "Score": 0.9961120519515884,
        "Text": "your autopay settings"
        },
        {
        "BeginOffset": 45,
        "EndOffset": 78,
        "Score": 0.9980620070116009,
        "Text": "your account Internet.org account"
        },
        {
        "BeginOffset": 97,
        "EndOffset": 109,
        "Score": 0.999919660140754,
        "Text": "your payment"
        },
        {
        "BeginOffset": 113,
        "EndOffset": 125,
        "Score": 0.9998370719754205,
        "Text": "the due date"
        },
        {
        "BeginOffset": 131,
        "EndOffset": 166,
        "Score": 0.9955068678502509,
        "Text": "your bank account number XXXXXX1111"
        },
        {
        "BeginOffset": 172,
        "EndOffset": 200,
        "Score": 0.8653433315829526,
        "Text": "the routing number XXXXX0000"
        }
    ],
    "Line": 0
    }
    {
```

```
    "File": "SampleText3.txt",
    "KeyPhrases": [
        {
        "BeginOffset": 0,
        "EndOffset": 4,
        "Score": 0.9142947833681668,
        "Text": "Jane"
        },
        {
        "BeginOffset": 20,
        "EndOffset": 41,
        "Score": 0.9984325676596763,
        "Text": "any customer feedback"
        },
        {
        "BeginOffset": 47,
        "EndOffset": 59,
        "Score": 0.9998782448150636,
        "Text": "this weekend"
        },
        {
        "BeginOffset": 63,
        "EndOffset": 75,
        "Score": 0.99866741830757,
        "Text": "Sunshine Spa"
        },
        {
        "BeginOffset": 77,
        "EndOffset": 88,
        "Score": 0.9695803485466054,
        "Text": "123 Main St"
        },
        {
        "BeginOffset": 108,
        "EndOffset": 116,
        "Score": 0.9997065928550928,
        "Text": "comments"
        },
        {
        "BeginOffset": 120,
        "EndOffset": 125,
        "Score": 0.9993466833825161,
        "Text": "Alice"
        },
```

```
        {
        "BeginOffset": 129,
        "EndOffset": 144,
        "Score": 0.9654563612885667,
        "Text": "AnySpa@example.com"
        }
    ],
    "Line": 0
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartKeyPhrasesDetectionJobà la section Référence des AWS CLI commandes.

`start-pii-entities-detection-job`

L'exemple de code suivant montre comment utiliser`start-pii-entities-detection-job`.

AWS CLI

Pour démarrer une tâche de détection d'informations personnelles asynchrone

L'`start-pii-entities-detection-job`exemple suivant lance une tâche de détection d'entités d'informations personnelles identifiables (PII) asynchrones pour tous les fichiers situés à l'adresse spécifiée par la balise. `--input-data-config` Dans cet exemple, le compartiment S3 contient `Sampletext1.txtSampletext2.txt`, et`Sampletext3.txt`. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config` balise. `output` Le dossier contient `SampleText1.txt.outSampleText2.txt.out`, et `SampleText3.txt.out` qui répertorient les entités nommées dans chaque fichier texte. La sortie Json est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-pii-entities-detection-job \
    --job-name entities_test \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
```

```
    --language-code en \
    --mode ONLY_OFFSETS
```

Contenu de `Sampletext1.txt` :

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st."
```

Contenu de `Sampletext2.txt` :

```
"Dear Max, based on your autopay settings for your account Internet.org account, we
 will withdraw your payment on the due date from your bank account number XXXXXX1111
 with the routing number XXXXX0000. "
```

Contenu de `Sampletext3.txt` :

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123
 Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu `SampleText1.txt.out` avec des retraits de ligne pour plus de lisibilité :

```
{
    "Entities": [
        {
        "BeginOffset": 6,
        "EndOffset": 15,
        "Type": "NAME",
        "Score": 0.9998490510222595
        },
        {
```

```
        "BeginOffset": 22,
        "EndOffset": 26,
        "Type": "NAME",
        "Score": 0.9998937958019426
        },
        {
        "BeginOffset": 88,
        "EndOffset": 107,
        "Type": "CREDIT_DEBIT_NUMBER",
        "Score": 0.9554297245278491
        },
        {
        "BeginOffset": 155,
        "EndOffset": 164,
        "Type": "DATE_TIME",
        "Score": 0.9999720462925257
        }
    ],
    "File": "SampleText1.txt",
    "Line": 0
}
```

Contenu `SampleText2.txt.out` avec des retraits de ligne pour plus de lisibilité :

```
{
    "Entities": [
        {
        "BeginOffset": 5,
        "EndOffset": 8,
        "Type": "NAME",
        "Score": 0.9994390774924007
        },
        {
        "BeginOffset": 58,
        "EndOffset": 70,
        "Type": "URL",
        "Score": 0.9999958276922101
        },
        {
        "BeginOffset": 156,
        "EndOffset": 166,
        "Type": "BANK_ACCOUNT_NUMBER",
        "Score": 0.9999721058045592
```

```
        },
        {
        "BeginOffset": 191,
        "EndOffset": 200,
        "Type": "BANK_ROUTING",
        "Score": 0.9998968945989909
        }
    ],
    "File": "SampleText2.txt",
    "Line": 0
}
```

Contenu `SampleText3.txt.out` avec des retraits de ligne pour plus de lisibilité :

```
{
    "Entities": [
        {
        "BeginOffset": 0,
        "EndOffset": 4,
        "Type": "NAME",
        "Score": 0.999949934606805
        },
        {
        "BeginOffset": 77,
        "EndOffset": 88,
        "Type": "ADDRESS",
        "Score": 0.9999035300466904
        },
        {
        "BeginOffset": 120,
        "EndOffset": 125,
        "Type": "NAME",
        "Score": 0.9998203838716296
        },
        {
        "BeginOffset": 129,
        "EndOffset": 144,
        "Type": "EMAIL",
        "Score": 0.9998313473105228
        }
    ],
    "File": "SampleText3.txt",
    "Line": 0
```

```
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartPiiEntitiesDetectionJobà la section Référence des AWS CLI commandes.

`start-sentiment-detection-job`

L'exemple de code suivant montre comment utiliser`start-sentiment-detection-job`.

AWS CLI

Pour démarrer une tâche d'analyse des sentiments asynchrone

L'`start-sentiment-detection-job`exemple suivant lance une tâche de détection d'analyse asynchrone des sentiments pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le dossier du compartiment S3 contient `SampleMovieReview1.txtSampleMovieReview2.txt`, et`SampleMovieReview3.txt`. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--output-data-config` balise. `output` Le dossier contient le fichier`output.txt`, qui contient les sentiments dominants pour chaque fichier texte et le score de confiance du modèle préentraîné pour chaque prédiction. La sortie Json est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-sentiment-detection-job \
    --job-name example-sentiment-detection-job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/MovieData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role
```

Contenu de `SampleMovieReview1.txt` :

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

Contenu de `SampleMovieReview2.txt` :

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I
 highly recommend this movie."
```

Contenu de `SampleMovieReview3.txt` :

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly
 stolen from other modern directors."
```

Sortie :

```
{
    "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobStatus": "SUBMITTED"
}
```

Contenu de `output.txt` avec ligne de retraits pour plus de lisibilité :

```
{
    "File": "SampleMovieReview1.txt",
        "Line": 0,
        "Sentiment": "MIXED",
        "SentimentScore": {
            "Mixed": 0.6591159105300903,
            "Negative": 0.26492202281951904,
            "Neutral": 0.035430654883384705,
            "Positive": 0.04053137078881264
            }
        }
    {
    "File": "SampleMovieReview2.txt",
        "Line": 0,
        "Sentiment": "POSITIVE",
        "SentimentScore": {
            "Mixed": 0.000008718466233403888,
            "Negative": 0.00006134175055194646,
            "Neutral": 0.0002941041602753103,
            "Positive": 0.9996358156204224
            }
        }
    {
```

```
    "File": "SampleMovieReview3.txt",
        "Line": 0,
        "Sentiment": "NEGATIVE",
        "SentimentScore": {
            "Mixed": 0.004146667663007975,
            "Negative": 0.9645107984542847,
            "Neutral": 0.016559595242142677,
            "Positive": 0.014782938174903393
        }
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartSentimentDetectionJobà la section Référence des AWS CLI commandes.

`start-targeted-sentiment-detection-job`

L'exemple de code suivant montre comment utiliser`start-targeted-sentiment-detection-job`.

AWS CLI

Pour démarrer une tâche d'analyse ciblée asynchrone des sentiments

L'`start-targeted-sentiment-detection-job`exemple suivant lance une tâche de détection d'analyse ciblée asynchrone des sentiments pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le dossier du compartiment S3 contient `SampleMovieReview1.txtSampleMovieReview2.txt`, et`SampleMovieReview3.txt`. Lorsque le travail est terminé, `output.tar.gz` il est placé à l'emplacement spécifié par le `--output-data-config` tag. `output.tar.gz`contient les fichiers`SampleMovieReview1.txt.out`,`SampleMovieReview2.txt.out`, et`SampleMovieReview3.txt.out`, qui contiennent chacun toutes les entités nommées et les sentiments associés pour un seul fichier texte d'entrée.

```
aws comprehend start-targeted-sentiment-detection-job \
    --job-name targeted_movie_review_analysis1 \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/MovieData" \
```

```
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role
```

Contenu de `SampleMovieReview1.txt` :

```
"The film, AnyMovie, is fairly predictable and just okay."
```

Contenu de `SampleMovieReview2.txt` :

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I
 highly recommend this movie."
```

Contenu de `SampleMovieReview3.txt` :

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly
 stolen from other modern directors."
```

Sortie :

```
{
    "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobStatus": "SUBMITTED"
}
```

Contenu `SampleMovieReview1.txt.out` avec des retraits de ligne pour plus de lisibilité :

```
{
    "Entities": [
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 4,
            "EndOffset": 8,
            "Score": 0.994972,
            "GroupScore": 1,
```

```
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 10,
            "EndOffset": 18,
            "Score": 0.631368,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "ORGANIZATION",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0.001729,
                "Negative": 0.000001,
                "Neutral": 0.000318,
                "Positive": 0.997952
                }
            }
            }
        ]
        }
    ],
    "File": "SampleMovieReview1.txt",
    "Line": 0
}
```

Contenu des retraits de `SampleMovieReview2.txt.out` ligne pour plus de lisibilité :


```
{
    "Entities": [
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 0,
            "EndOffset": 8,
            "Score": 0.854024,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 0.000007,
                "Positive": 0.999993
                }
            }
            },
            {
            "BeginOffset": 104,
            "EndOffset": 109,
            "Score": 0.999129,
            "GroupScore": 0.502937,
            "Text": "movie",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 0,
                "Positive": 1
                }
            }
            },
            {
```

```
            "BeginOffset": 33,
            "EndOffset": 37,
            "Score": 0.999823,
            "GroupScore": 0.999252,
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 0.000001,
                "Positive": 0.999999
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0,
            1,
            2
        ],
        "Mentions": [
            {
            "BeginOffset": 43,
            "EndOffset": 44,
            "Score": 0.999997,
            "GroupScore": 1,
            "Text": "I",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            },
            {
            "BeginOffset": 80,
```

```
            "EndOffset": 81,
            "Score": 0.999996,
            "GroupScore": 0.52523,
            "Text": "I",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            },
            {
            "BeginOffset": 67,
            "EndOffset": 68,
            "Score": 0.999994,
            "GroupScore": 0.999499,
            "Text": "I",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 75,
            "EndOffset": 78,
            "Score": 0.999978,
            "GroupScore": 1,
```

```
            "Text": "kid",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
        }
    ],
    "File": "SampleMovieReview2.txt",
    "Line": 0
}
```

Contenu `SampleMovieReview3.txt.out` avec des retraits de ligne pour plus de lisibilité :

```
{
    "Entities": [
        {
        "DescriptiveMentionIndex": [
            1
        ],
        "Mentions": [
            {
            "BeginOffset": 64,
            "EndOffset": 68,
            "Score": 0.992953,
            "GroupScore": 0.999814,
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0.000004,
                "Negative": 0.010425,
                "Neutral": 0.989543,
                "Positive": 0.000027
                }
```

```
            }
            },
            {
            "BeginOffset": 37,
            "EndOffset": 45,
            "Score": 0.999782,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "ORGANIZATION",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0.000095,
                "Negative": 0.039847,
                "Neutral": 0.000673,
                "Positive": 0.959384
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 47,
            "EndOffset": 50,
            "Score": 0.999991,
            "GroupScore": 1,
            "Text": "All",
            "Type": "QUANTITY",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0.000001,
                "Negative": 0.000001,
                "Neutral": 0.999998,
                "Positive": 0
                }
            }
            }
        ]
```

```
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 106,
            "EndOffset": 115,
            "Score": 0.542083,
            "GroupScore": 1,
            "Text": "directors",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
        }
    ],
    "File": "SampleMovieReview3.txt",
    "Line": 0
}
```


Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartTargetedSentimentDetectionJobà la section Référence des AWS CLI commandes.

`start-topics-detection-job`

L'exemple de code suivant montre comment utiliser`start-topics-detection-job`.

AWS CLI

Pour démarrer une tâche d'analyse de détection de sujets

L'`start-topics-detection-job`exemple suivant lance une tâche de détection de sujets asynchrones pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--ouput-data-config` balise. `output` `output`contient topic-terms.csv et doc-topics.csv. Le premier fichier de sortie, topic-terms.csv, est une liste des rubriques de la collection. Pour chaque sujet, la liste inclut, par défaut, les principaux termes par sujet en fonction de leur poids. Le second fichier répertorie `doc-topics.csv` les documents associés à un sujet et la proportion du document qui traite du sujet.

```
aws comprehend start-topics-detection-job \
    --job-name example_topics_detection_job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Pour plus d'informations, consultez la section Modélisation des rubriques dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartTopicsDetectionJobà la section Référence des AWS CLI commandes.

`stop-dominant-language-detection-job`

L'exemple de code suivant montre comment utiliser`stop-dominant-language-detection-job`.

AWS CLI

Pour arrêter une tâche de détection de langue dominante asynchrone

L'`stop-dominant-language-detection-job`exemple suivant arrête une tâche de détection de langue dominante asynchrone en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopDominantLanguageDetectionJobà la section Référence des AWS CLI commandes.

`stop-entities-detection-job`

L'exemple de code suivant montre comment utiliser`stop-entities-detection-job`.

AWS CLI

Pour arrêter une tâche de détection d'entités asynchrones

L'`stop-entities-detection-job`exemple suivant arrête une tâche de détection d'entités asynchrones en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
```

```
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopEntitiesDetectionJobà la section Référence des AWS CLI commandes.

`stop-events-detection-job`

L'exemple de code suivant montre comment utiliser`stop-events-detection-job`.

AWS CLI

Pour arrêter une tâche de détection d'événements asynchrones

L'`stop-events-detection-job`exemple suivant arrête une tâche de détection d'événements asynchrones en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-events-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopEventsDetectionJobà la section Référence des AWS CLI commandes.

`stop-key-phrases-detection-job`

L'exemple de code suivant montre comment utiliser`stop-key-phrases-detection-job`.

AWS CLI

Pour arrêter une tâche de détection de phrases-clés asynchrones

L'`stop-key-phrases-detection-job`exemple suivant arrête une tâche de détection de phrases clés asynchrones en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopKeyPhrasesDetectionJobà la section Référence des AWS CLI commandes.

`stop-pii-entities-detection-job`

L'exemple de code suivant montre comment utiliser`stop-pii-entities-detection-job`.

AWS CLI

Pour arrêter une tâche de détection d'entités pii asynchrones

L'`stop-pii-entities-detection-job`exemple suivant arrête une tâche de détection d'entités pii asynchrones en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopPiiEntitiesDetectionJobà la section Référence des AWS CLI commandes.

`stop-sentiment-detection-job`

L'exemple de code suivant montre comment utiliser`stop-sentiment-detection-job`.

AWS CLI

Pour arrêter une tâche de détection de sentiments asynchrone

L'`stop-sentiment-detection-job`exemple suivant arrête une tâche asynchrone de détection des sentiments en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopSentimentDetectionJobà la section Référence des AWS CLI commandes.

`stop-targeted-sentiment-detection-job`

L'exemple de code suivant montre comment utiliser`stop-targeted-sentiment-detection-job`.

AWS CLI

Pour arrêter une tâche de détection de sentiments ciblée asynchrone

L'`stop-targeted-sentiment-detection-job`exemple suivant arrête une tâche de détection de sentiments ciblée asynchrone en cours. Si l'état actuel du travail est `IN_PROGRESS` le suivant, le travail est marqué pour fin et placé dans `STOP_REQUESTED` cet état. Si la tâche se termine avant de pouvoir être arrêtée, elle est mise en `COMPLETED` état.

```
aws comprehend stop-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopTargetedSentimentDetectionJobà la section Référence des AWS CLI commandes.

`stop-training-document-classifier`

L'exemple de code suivant montre comment utiliser`stop-training-document-classifier`.

AWS CLI

Pour arrêter l'entraînement d'un modèle de classificateur de documents

L'`stop-training-document-classifier`exemple suivant arrête l'entraînement d'un modèle de classificateur de documents alors qu'il est en cours.

```
aws comprehend stop-training-document-classifier
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopTrainingDocumentClassifierà la section Référence des AWS CLI commandes.

`stop-training-entity-recognizer`

L'exemple de code suivant montre comment utiliser`stop-training-entity-recognizer`.

AWS CLI

Pour arrêter la formation d'un modèle de reconnaissance d'entités

L'`stop-training-entity-recognizer`exemple suivant arrête l'entraînement d'un modèle de reconnaissance d'entités alors qu'il est en cours.

```
aws comprehend stop-training-entity-recognizer
    --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/examplerecognizer1"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopTrainingEntityRecognizerà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Exemple 1 : pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute une seule balise à une ressource Amazon Comprehend.

```
aws comprehend tag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
    --tags Key=Location,Value=Seattle
```

Cette commande n'a aucune sortie.

Pour plus d'informations, consultez la section Marquage de vos ressources dans le manuel Amazon Comprehend Developer Guide.

Exemple 2 : pour ajouter plusieurs balises à une ressource

L'`tag-resource`exemple suivant ajoute plusieurs balises à une ressource Amazon Comprehend.

```
aws comprehend tag-resource \
    --resource-arn "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1" \
    --tags Key=location,Value=Seattle Key=Department,Value=Finance
```

Cette commande n'a aucune sortie.

Pour plus d'informations, consultez la section Marquage de vos ressources dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Exemple 1 : pour supprimer une seule balise d'une ressource

L'`untag-resource`exemple suivant supprime une seule balise d'une ressource Amazon Comprehend.

```
aws comprehend untag-resource \
```

```
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
    --tag-keys Location
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources dans le manuel Amazon Comprehend Developer Guide.

Exemple 2 : pour supprimer plusieurs balises d'une ressource

L'`untag-resource`exemple suivant supprime plusieurs balises d'une ressource Amazon Comprehend.

```
aws comprehend untag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
    --tag-keys Location Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-endpoint`

L'exemple de code suivant montre comment utiliser`update-endpoint`.

AWS CLI

Exemple 1 : pour mettre à jour les unités d'inférence d'un point de terminaison

L'`update-endpoint`exemple suivant met à jour les informations relatives à un point de terminaison. Dans cet exemple, le nombre d'unités d'inférence est augmenté.

```
aws comprehend update-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint
    --desired-inference-units 2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

Exemple 2 : pour mettre à jour le modèle actif d'un point de terminaison

L'`update-endpoint`exemple suivant met à jour les informations relatives à un point de terminaison. Dans cet exemple, le modèle actif est modifié.

```
aws comprehend update-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint
    --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-new
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateEndpointà la section Référence des AWS CLI commandes.

`update-flywheel`

L'exemple de code suivant montre comment utiliser`update-flywheel`.

AWS CLI

Pour mettre à jour la configuration d'un volant

L'`update-flywheel`exemple suivant met à jour la configuration d'un volant d'inertie. Dans cet exemple, le modèle actif du volant est mis à jour.

```
aws comprehend update-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1 \
    --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/new-example-classifier-model
```

Sortie :

```
{
    "FlywheelProperties": {
        "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity",
        "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/new-example-classifier-model",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "TaskConfig": {
            "LanguageCode": "en",
            "DocumentClassificationConfig": {
                "Mode": "MULTI_CLASS"
            }
        },
        "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/
schemaVersion=1/20230616T200543Z/",
        "DataSecurityConfig": {},
        "Status": "ACTIVE",
        "ModelType": "DOCUMENT_CLASSIFIER",
        "CreationTime": "2023-06-16T20:05:43.242000+00:00",
        "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
        "LatestFlywheelIteration": "20230619T040032Z"
    }
}
```

Pour plus d'informations, consultez la présentation de Flywheel dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateFlywheelà la section Référence des AWS CLI commandes.

Exemples d'Amazon Comprehend Medical utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Comprehend Medical.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-entities-detection-v2-job`

L'exemple de code suivant montre comment utiliser`describe-entities-detection-v2-job`.

AWS CLI

Pour décrire une tâche de détection d'entités

L'`describe-entities-detection-v2-job`exemple suivant affiche les propriétés associées à une tâche de détection d'entités asynchrones.

```
aws comprehendmedical describe-entities-detection-v2-job \
    --job-id "ab9887877365fe70299089371c043b96"
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "ab9887877365fe70299089371c043b96",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-03-18T21:20:15.614000+00:00",
        "EndTime": "2020-03-18T21:27:07.350000+00:00",
        "ExpirationTime": "2020-07-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": ""
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "DetectEntitiesModelV20190930"
    }
```

```
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, consultez DescribeEntitiesDetectionV2Job dans AWS CLI Command Reference.

`describe-icd10-cm-inference-job`

L'exemple de code suivant montre comment utiliser`describe-icd10-cm-inference-job`.

AWS CLI

Pour décrire une tâche d'inférence ICD-10-CM

L'`describe-icd10-cm-inference-job`exemple suivant décrit les propriétés de la tâche d'inférence demandée avec l'identifiant de tâche spécifié.

```
aws comprehendmedical describe-icd10-cm-inference-job \
    --job-id "5780034166536cdb52ffa3295a1b00a7"
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "5780034166536cdb52ffa3295a1b00a7",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
        "EndTime": "2020-05-18T21:27:07.350000+00:00",
        "ExpirationTime": "2020-09-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
```

```
        "ModelVersion":  "0.1.0"
    }
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous à la section DescribeIcd10 CmInferenceJob de AWS CLI la section Référence des commandes.

`describe-phi-detection-job`

L'exemple de code suivant montre comment utiliser`describe-phi-detection-job`.

AWS CLI

Pour décrire une tâche de détection de PHI

L'`describe-phi-detection-job`exemple suivant affiche les propriétés associées à une tâche de détection asynchrone d'informations de santé protégées (PHI).

```
aws comprehendmedical describe-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "4750034166536cdb52ffa3295a1b00a3",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
        "EndTime": "2020-03-19T20:45:07.894000+00:00",
        "ExpirationTime": "2020-07-17T20:38:37+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": ""
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "867139942017-PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
        },
        "LanguageCode": "en",
```

```
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "PHIModelV20190903"
    }
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribePhiDetectionJobà la section Référence des AWS CLI commandes.

`describe-rx-norm-inference-job`

L'exemple de code suivant montre comment utiliser`describe-rx-norm-inference-job`.

AWS CLI

Pour décrire une tâche d' RxNorm inférence

L'`describe-rx-norm-inference-job`exemple suivant décrit les propriétés de la tâche d'inférence demandée avec l'identifiant de tâche spécifié.

```
aws comprehendmedical describe-rx-norm-inference-job \
    --job-id "eg8199877365fc70299089371c043b96"
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "g8199877365fc70299089371c043b96",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
        "EndTime": "2020-05-18T21:27:07.350000+00:00",
        "ExpirationTime": "2020-09-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
```

```
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "0.0.0"
    }
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeRxNormInferenceJobà la section Référence des AWS CLI commandes.

`describe-snomedct-inference-job`

L'exemple de code suivant montre comment utiliser`describe-snomedct-inference-job`.

AWS CLI

Pour décrire une tâche d'inférence SNOMED CT

L'`describe-snomedct-inference-job`exemple suivant décrit les propriétés de la tâche d'inférence demandée avec l'identifiant de tâche spécifié.

```
aws comprehendmedical describe-snomedct-inference-job \
    --job-id "2630034166536cdb52ffa3295a1b00a7"
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "2630034166536cdb52ffa3295a1b00a7",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2021-12-18T21:20:15.614000+00:00",
        "EndTime": "2021-12-18T21:27:07.350000+00:00",
        "ExpirationTime": "2022-05-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "OutputDataConfig": {
```

```
            "S3Bucket": "comp-med-output",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion":  "0.1.0"
    }
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeSnomedctInferenceJobà la section Référence des AWS CLI commandes.

`detect-entities-v2`

L'exemple de code suivant montre comment utiliser`detect-entities-v2`.

AWS CLI

Exemple 1 : pour détecter des entités directement à partir du texte

L'`detect-entities-v2`exemple suivant montre les entités détectées et les étiquette en fonction de leur type, directement à partir du texte saisi.

```
aws comprehendmedical detect-entities-v2 \
    --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and
 leg, slightly itchy."
```

Sortie :

```
{
    "Id": 0,
    "BeginOffset": 38,
    "EndOffset": 47,
    "Score": 0.9942955374717712,
    "Text": "Clonidine",
    "Category": "MEDICATION",
    "Type": "GENERIC_NAME",
    "Traits": []
```

```
}
```

Pour plus d'informations, consultez la version 2 de Detect Entities dans le manuel Amazon Comprehend Medical Developer Guide.

Exemple 2 : pour détecter des entités à partir d'un chemin de fichier

L'`detect-entities-v2`exemple suivant montre les entités détectées et les étiquette en fonction de leur type à partir d'un chemin de fichier.

```
aws comprehendmedical detect-entities-v2 \
    --text file://medical_entities.txt
```

Contenu de `medical_entities.txt` :

```
{
    "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,
 slightly itchy."
}
```

Sortie :

```
{
    "Id": 0,
    "BeginOffset": 38,
    "EndOffset": 47,
    "Score": 0.9942955374717712,
    "Text": "Clonidine",
    "Category": "MEDICATION",
    "Type": "GENERIC_NAME",
    "Traits": []
}
```

Pour plus d'informations, consultez la version 2 de Detect Entities dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir DetectEntitiesV2 dans AWS CLI Command Reference.

`detect-phi`

L'exemple de code suivant montre comment utiliser`detect-phi`.

AWS CLI

Exemple 1 : pour détecter les informations de santé protégées (PHI) directement à partir du texte

L'`detect-phi`exemple suivant affiche les entités d'informations de santé protégées (PHI) détectées directement à partir du texte saisi.

```
aws comprehendmedical detect-phi \
    --text "Patient Carlos Salazar presented with rash on his upper extremities and dry cough. He lives at 100 Main Street, Anytown, USA where he works from his home as a carpenter."
```

Sortie :

```
{
    "Entities": [
        {
            "Id": 0,
            "BeginOffset": 8,
            "EndOffset": 21,
            "Score": 0.9914507269859314,
            "Text": "Carlos Salazar",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "NAME",
            "Traits": []
        },
        {
            "Id": 1,
            "BeginOffset": 94,
            "EndOffset": 109,
            "Score": 0.871849775314331,
            "Text": "100 Main Street, Anytown, USA",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "ADDRESS",
            "Traits": []
        },
        {
            "Id": 2,
            "BeginOffset": 145,
            "EndOffset": 154,
            "Score": 0.8302185535430908,
            "Text": "carpenter",
            "Category": "PROTECTED_HEALTH_INFORMATION",
```

```
            "Type": "PROFESSION",
            "Traits": []
        }
    ],
    "ModelVersion": "0.0.0"
}
```

Pour plus d'informations, consultez Detect PHI dans le manuel Amazon Comprehend Medical Developer Guide.

Exemple 2 : pour détecter et protéger les informations de santé (PHI) directement à partir d'un chemin de fichier

L'`detect-phi`exemple suivant montre les entités d'informations de santé protégées (PHI) détectées à partir d'un chemin de fichier.

```
aws comprehendmedical detect-phi \
    --text file://phi.txt
```

Contenu de `phi.txt` :

```
"Patient Carlos Salazar presented with a rash on his upper extremities and a dry
 cough. He lives at 100 Main Street, Anytown, USA, where he works from his home as a
 carpenter."
```

Sortie :

```
{
    "Entities": [
        {
            "Id": 0,
            "BeginOffset": 8,
            "EndOffset": 21,
            "Score": 0.9914507269859314,
            "Text": "Carlos Salazar",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "NAME",
            "Traits": []
        },
        {
            "Id": 1,
            "BeginOffset": 94,
```

```
            "EndOffset": 109,
            "Score": 0.871849775314331,
            "Text": "100 Main Street, Anytown, USA",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "ADDRESS",
            "Traits": []
        },
        {
            "Id": 2,
            "BeginOffset": 145,
            "EndOffset": 154,
            "Score": 0.8302185535430908,
            "Text": "carpenter",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "PROFESSION",
            "Traits": []
        }
    ],
    "ModelVersion": "0.0.0"
}
```

Pour plus d'informations, consultez Detect PHI dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir DetectPhila section Référence des AWS CLI commandes.

`infer-icd10-cm`

L'exemple de code suivant montre comment utiliser`infer-icd10-cm`.

AWS CLI

Exemple 1 : pour détecter des entités liées à un état médical et créer un lien vers l'ontologie ICD-10-CM directement à partir du texte

L'`infer-icd10-cm`exemple suivant étiquette les entités des affections médicales détectées et associe ces entités aux codes de l'édition 2019 de la Classification internationale des maladies, modification clinique (ICD-10-CM).

```
aws comprehendmedical infer-icd10-cm \
    --text "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

Sortie :


```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "abdominal pain",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9475538730621338,
            "BeginOffset": 28,
            "EndOffset": 42,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "SYMPTOM",
                    "Score": 0.6724207401275635
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Unspecified abdominal pain",
                    "Code": "R10.9",
                    "Score": 0.6904221177101135
                },
                {
                    "Description": "Epigastric pain",
                    "Code": "R10.13",
                    "Score": 0.1364113688468933
                },
                {
                    "Description": "Generalized abdominal pain",
                    "Code": "R10.84",
                    "Score": 0.12508003413677216
                },
                {
                    "Description": "Left lower quadrant pain",
                    "Code": "R10.32",
                    "Score": 0.10063883662223816
                },
                {
                    "Description": "Lower abdominal pain, unspecified",
                    "Code": "R10.30",
                    "Score": 0.09933677315711975
```

```
                }
            ]
        },
        {
            "Id": 1,
            "Text": "diabetes",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9899052977561951,
            "BeginOffset": 75,
            "EndOffset": 83,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "DIAGNOSIS",
                    "Score": 0.9258432388305664
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Type 2 diabetes mellitus without complications",
                    "Code": "E11.9",
                    "Score": 0.7158446311950684
                },
                {
                    "Description": "Family history of diabetes mellitus",
                    "Code": "Z83.3",
                    "Score": 0.5704703330993652
                },
                {
                    "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
                    "Code": "Z83.49",
                    "Score": 0.19856023788452148
                },
                {
                    "Description": "Type 1 diabetes mellitus with ketoacidosis
without coma",
                    "Code": "E10.10",
                    "Score": 0.13285516202449799
                },
                {
                    "Description": "Type 2 diabetes mellitus with hyperglycemia",
                    "Code": "E11.65",
```

```
                    "Score": 0.0993388369679451
                }
            ]
        }
    ],
    "ModelVersion": "0.1.0"
}
```

Pour plus d'informations, consultez Infer ICD1 0-CM dans le manuel Amazon Comprehend Medical Developer Guide.

Exemple 2 : pour détecter des entités liées à un état médical et établir un lien vers l'ontologie ICD-10-CM à partir d'un chemin de fichier

L'`infer-icd-10-cm`exemple suivant étiquette les entités des affections médicales détectées et associe ces entités aux codes de l'édition 2019 de la Classification internationale des maladies, modification clinique (ICD-10-CM).

```
aws comprehendmedical infer-icd10-cm \
    --text file://icd10cm.txt
```

Contenu de `icd10cm.txt` :

```
{
    "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
}
```

Sortie :

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "abdominal pain",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9475538730621338,
            "BeginOffset": 28,
            "EndOffset": 42,
            "Attributes": [],
            "Traits": [
```

```
                {
                    "Name": "SYMPTOM",
                    "Score": 0.6724207401275635
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Unspecified abdominal pain",
                    "Code": "R10.9",
                    "Score": 0.6904221177101135
                },
                {
                    "Description": "Epigastric pain",
                    "Code": "R10.13",
                    "Score": 0.1364113688468933
                },
                {
                    "Description": "Generalized abdominal pain",
                    "Code": "R10.84",
                    "Score": 0.12508003413677216
                },
                {
                    "Description": "Left lower quadrant pain",
                    "Code": "R10.32",
                    "Score": 0.10063883662223816
                },
                {
                    "Description": "Lower abdominal pain, unspecified",
                    "Code": "R10.30",
                    "Score": 0.09933677315711975
                }
            ]
        },
        {
            "Id": 1,
            "Text": "diabetes",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9899052977561951,
            "BeginOffset": 75,
            "EndOffset": 83,
            "Attributes": [],
            "Traits": [
                {
```

```
                    "Name": "DIAGNOSIS",
                    "Score": 0.9258432388305664
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Type 2 diabetes mellitus without complications",
                    "Code": "E11.9",
                    "Score": 0.7158446311950684
                },
                {
                    "Description": "Family history of diabetes mellitus",
                    "Code": "Z83.3",
                    "Score": 0.5704703330993652
                },
                {
                    "Description": "Family history of other endocrine, nutritional
 and metabolic diseases",
                    "Code": "Z83.49",
                    "Score": 0.19856023788452148
                },
                {
                    "Description": "Type 1 diabetes mellitus with ketoacidosis
 without coma",
                    "Code": "E10.10",
                    "Score": 0.13285516202449799
                },
                {
                    "Description": "Type 2 diabetes mellitus with hyperglycemia",
                    "Code": "E11.65",
                    "Score": 0.0993388369679451
                }
            ]
        }
    ],
    "ModelVersion": "0.1.0"
}
```


Pour plus d'informations, consultez ICD1Infer-0-CM dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir InferIcd10Cm dans AWS CLI Command Reference.

`infer-rx-norm`

L'exemple de code suivant montre comment utiliser`infer-rx-norm`.

AWS CLI

Exemple 1 : pour détecter les entités médicamenteuses et créer un lien vers celles-ci RxNorm directement à partir du texte

L'`infer-rx-norm`exemple suivant montre et étiquette les entités médicamenteuses détectées et lie ces entités aux identificateurs conceptuels (RxCUI) de la base de données de la National Library of Medicine. RxNorm

```
aws comprehendmedical infer-rx-norm \
    --text "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but
 denies taking Synthroid."
```

Sortie :

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "Levothyroxine",
            "Category": "MEDICATION",
            "Type": "GENERIC_NAME",
            "Score": 0.9996285438537598,
            "BeginOffset": 23,
            "EndOffset": 36,
            "Attributes": [
                {
                    "Type": "DOSAGE",
                    "Score": 0.9892290830612183,
                    "RelationshipScore": 0.9997978806495667,
                    "Id": 1,
                    "BeginOffset": 37,
                    "EndOffset": 51,
                    "Text": "125 micrograms",
                    "Traits": []
                },
                {
                    "Type": "ROUTE_OR_MODE",
                    "Score": 0.9988924860954285,
```

```
                    "RelationshipScore": 0.998291552066803,
                    "Id": 2,
                    "BeginOffset": 52,
                    "EndOffset": 56,
                    "Text": "p.o.",
                    "Traits": []
                },
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9953463673591614,
                    "RelationshipScore": 0.9999889135360718,
                    "Id": 3,
                    "BeginOffset": 57,
                    "EndOffset": 67,
                    "Text": "once daily",
                    "Traits": []
                }
            ],
            "Traits": [],
            "RxNormConcepts": [
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
                    "Code": "966224",
                    "Score": 0.9912070631980896
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
                    "Code": "966405",
                    "Score": 0.8698278665542603
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.7448257803916931
                },
                {
                    "Description": "levothyroxine",
                    "Code": "10582",
                    "Score": 0.7050482630729675
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
```

```
                    "Code": "966190",
                    "Score": 0.6921631693840027
                }
            ]
        },
        {
            "Id": 4,
            "Text": "Synthroid",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Score": 0.9946461319923401,
            "BeginOffset": 86,
            "EndOffset": 95,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "NEGATION",
                    "Score": 0.5167351961135864
                }
            ],
            "RxNormConcepts": [
                {
                    "Description": "Synthroid",
                    "Code": "224920",
                    "Score": 0.9462039470672607
                },
                {
                    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
                    "Code": "966282",
                    "Score": 0.8309829235076904
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.4945160448551178
                },
                {
                    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
[Synthroid]",
                    "Code": "966247",
                    "Score": 0.3674522042274475
                },
```

```
                {
                    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
 [Synthroid]",
                    "Code": "966158",
                    "Score": 0.2588822841644287
                }
            ]
        }
    ],
    "ModelVersion": "0.0.0"
}
```

Pour plus d'informations, consultez la section Infer du RxNorm guide du développeur Amazon Comprehend Medical.

Exemple 2 : pour détecter des entités médicamenteuses et établir un lien vers celles-ci à RxNorm partir d'un chemin de fichier.

L'`infer-rx-norm`exemple suivant montre et étiquette les entités médicamenteuses détectées et lie ces entités aux identificateurs conceptuels (RxCUI) de la base de données de la National Library of Medicine. RxNorm

```
aws comprehendmedical infer-rx-norm \
    --text file://rxnorm.txt
```

Contenu de `rxnorm.txt` :

```
{
    "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies
 taking Synthroid."
}
```

Sortie :

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "Levothyroxine",
            "Category": "MEDICATION",
            "Type": "GENERIC_NAME",
            "Score": 0.9996285438537598,
```

```
            "BeginOffset": 23,
            "EndOffset": 36,
            "Attributes": [
                {
                    "Type": "DOSAGE",
                    "Score": 0.9892290830612183,
                    "RelationshipScore": 0.9997978806495667,
                    "Id": 1,
                    "BeginOffset": 37,
                    "EndOffset": 51,
                    "Text": "125 micrograms",
                    "Traits": []
                },
                {
                    "Type": "ROUTE_OR_MODE",
                    "Score": 0.9988924860954285,
                    "RelationshipScore": 0.998291552066803,
                    "Id": 2,
                    "BeginOffset": 52,
                    "EndOffset": 56,
                    "Text": "p.o.",
                    "Traits": []
                },
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9953463673591614,
                    "RelationshipScore": 0.9999889135360718,
                    "Id": 3,
                    "BeginOffset": 57,
                    "EndOffset": 67,
                    "Text": "once daily",
                    "Traits": []
                }
            ],
            "Traits": [],
            "RxNormConcepts": [
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
                    "Code": "966224",
                    "Score": 0.9912070631980896
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
                    "Code": "966405",
```

```
                    "Score": 0.8698278665542603
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.7448257803916931
                },
                {
                    "Description": "levothyroxine",
                    "Code": "10582",
                    "Score": 0.7050482630729675
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
                    "Code": "966190",
                    "Score": 0.6921631693840027
                }
            ]
        },
        {
            "Id": 4,
            "Text": "Synthroid",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Score": 0.9946461319923401,
            "BeginOffset": 86,
            "EndOffset": 95,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "NEGATION",
                    "Score": 0.5167351961135864
                }
            ],
            "RxNormConcepts": [
                {
                    "Description": "Synthroid",
                    "Code": "224920",
                    "Score": 0.9462039470672607
                },
                {
```

```
                    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
 [Synthroid]",
                    "Code": "966282",
                    "Score": 0.8309829235076904
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
 [Synthroid]",
                    "Code": "966191",
                    "Score": 0.4945160448551178
                },
                {
                    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
 [Synthroid]",
                    "Code": "966247",
                    "Score": 0.3674522042274475
                },
                {
                    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
 [Synthroid]",
                    "Code": "966158",
                    "Score": 0.2588822841644287
                }
            ]
        }
    ],
    "ModelVersion": "0.0.0"
}
```

Pour plus d'informations, consultez la section Infer du RxNorm guide du développeur Amazon Comprehend Medical.

- Pour plus de détails sur l'API, voir InferRxNormla section Référence des AWS CLI commandes.

`infer-snomedct`

L'exemple de code suivant montre comment utiliser`infer-snomedct`.

AWS CLI

Exemple : pour détecter des entités et établir un lien vers l'ontologie SNOMED CT directement à partir du texte

L'`infer-snomedct`exemple suivant montre comment détecter les entités médicales et les relier aux concepts de la version 2021-03 de la Nomenclature systématisée de la médecine, termes cliniques (SNOMED CT).

```
aws comprehendmedical infer-snomedct \
    --text "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

Sortie :

```
{
    "Entities": [
        {
            "Id": 3,
            "BeginOffset": 26,
            "EndOffset": 40,
            "Score": 0.9598260521888733,
            "Text": "abdominal pain",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Traits": [
                {
                    "Name": "SYMPTOM",
                    "Score": 0.6819021701812744
                }
            ]
        },
        {
            "Id": 4,
            "BeginOffset": 73,
            "EndOffset": 81,
            "Score": 0.9905840158462524,
            "Text": "diabetes",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Traits": [
                {
                    "Name": "DIAGNOSIS",
                    "Score": 0.9255214333534241
                }
            ]
        },
        {
```

```
            "Id": 1,
            "BeginOffset": 95,
            "EndOffset": 104,
            "Score": 0.6371926665306091,
            "Text": "Micronase",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Traits": [],
            "Attributes": [
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9761165380477905,
                    "RelationshipScore": 0.9984188079833984,
                    "RelationshipType": "FREQUENCY",
                    "Id": 2,
                    "BeginOffset": 105,
                    "EndOffset": 110,
                    "Text": "daily",
                    "Category": "MEDICATION",
                    "Traits": []
                }
            ]
        }
    ],
    "UnmappedAttributes": [],
    "ModelVersion": "1.0.0"
}
```


Pour plus d'informations, consultez InfersNoMedCT dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir InferSnomedctla section Référence des AWS CLI commandes.

`list-entities-detection-v2-jobs`

L'exemple de code suivant montre comment utiliser`list-entities-detection-v2-jobs`.

AWS CLI

Pour répertorier les tâches de détection d'entités

L'`list-entities-detection-v2-jobs`exemple suivant répertorie les tâches de détection asynchrones en cours.

```
aws comprehendmedical list-entities-detection-v2-jobs
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "ab9887877365fe70299089371c043b96",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
            "EndTime": "2020-03-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-07-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": ""
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion": "DetectEntitiesModelV20190930"
        }
    ]
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, consultez ListEntitiesDetectionV2Jobs dans AWS CLI Command Reference.

`list-icd10-cm-inference-jobs`

L'exemple de code suivant montre comment utiliser`list-icd10-cm-inference-jobs`.

AWS CLI

Pour répertorier tous les travaux d'inférence ICD-10-CM actuels

L'exemple suivant montre comment l'`list-icd10-cm-inference-jobs`opération renvoie une liste des tâches d'inférence par lots asynchrones ICD-10-CM en cours.

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "5780034166536cdb52ffa3295a1b00a7",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
            "EndTime": "2020-05-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-09-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion":  "0.1.0"
        }
    ]
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous à la section ListIcd10 CmInferenceJobs de AWS CLI la section Référence des commandes.

`list-phi-detection-jobs`

L'exemple de code suivant montre comment utiliser`list-phi-detection-jobs`.

AWS CLI

Pour répertorier les tâches de détection d'informations de santé protégées (PHI)

L'`list-phi-detection-jobs`exemple suivant répertorie les tâches de détection d'informations de santé protégées (PHI) en cours

```
aws comprehendmedical list-phi-detection-jobs
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "4750034166536cdb52ffa3295a1b00a3",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
            "EndTime": "2020-03-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-07-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": ""
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "867139942017-
PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion": "PHIModelV20190903"
        }
    ]
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir ListPhiDetectionJobsla section Référence des AWS CLI commandes.

`list-rx-norm-inference-jobs`

L'exemple de code suivant montre comment utiliser`list-rx-norm-inference-jobs`.

AWS CLI

Pour répertorier tous les travaux d'inférence Rx-Norm actuels

L'exemple suivant montre comment `list-rx-norm-inference-jobs` renvoie une liste des tâches d'inférence par lots Rx-Norm asynchrones en cours.

```
aws comprehendmedical list-rx-norm-inference-jobs
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "4980034166536cfb52gga3295a1b00a3",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
            "EndTime": "2020-05-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-09-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion": "0.0.0"
        }
    ]
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir ListRxNormInferenceJobsla section Référence des AWS CLI commandes.

`list-snomedct-inference-jobs`

L'exemple de code suivant montre comment utiliser`list-snomedct-inference-jobs`.

AWS CLI

Pour répertorier tous les jobs d'inférence SNOMED CT

L'exemple suivant montre comment l'`list-snomedct-inference-jobs`opération renvoie une liste des tâches d'inférence par lots asynchrones SNOMED CT en cours.

```
aws comprehendmedical list-snomedct-inference-jobs
```

Sortie :

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "5780034166536cdb52ffa3295a1b00a7",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
            "EndTime": "2020-05-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-09-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion":  "0.1.0"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir ListSnomedctInferenceJobsla section Référence des AWS CLI commandes.

`start-entities-detection-v2-job`

L'exemple de code suivant montre comment utiliser`start-entities-detection-v2-job`.

AWS CLI

Pour démarrer une tâche de détection d'entités

L'`start-entities-detection-v2-job`exemple suivant démarre une tâche de détection d'entités asynchrones.

```
aws comprehendmedical start-entities-detection-v2-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

Sortie :

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, consultez StartEntitiesDetectionV2Job dans AWS CLI Command Reference.

`start-icd10-cm-inference-job`

L'exemple de code suivant montre comment utiliser`start-icd10-cm-inference-job`.

AWS CLI

Pour démarrer une tâche d'inférence ICD-10-CM

L'`start-icd10-cm-inference-job`exemple suivant démarre une tâche d'analyse par lots d'inférence ICD-10-CM.

```
aws comprehendmedical start-icd10-cm-inference-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

Sortie :

```
{
    "JobId": "ef7289877365fc70299089371c043b96"
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous à la section StartIcd10 CmInferenceJob de AWS CLI la section Référence des commandes.

`start-phi-detection-job`

L'exemple de code suivant montre comment utiliser`start-phi-detection-job`.

AWS CLI

Pour démarrer une tâche de détection de PHI

L'`start-phi-detection-job`exemple suivant démarre une tâche de détection d'entités PHI asynchrones.

```
aws comprehendmedical start-phi-detection-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
```

```
    --language-code en
```

Sortie :

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir StartPhiDetectionJobla section Référence des AWS CLI commandes.

`start-rx-norm-inference-job`

L'exemple de code suivant montre comment utiliser`start-rx-norm-inference-job`.

AWS CLI

Pour démarrer une tâche d' RxNorm inférence

L'`start-rx-norm-inference-job`exemple suivant démarre une tâche d'analyse par lots d' RxNorm inférence.

```
aws comprehendmedical start-rx-norm-inference-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

Sortie :

```
{
    "JobId": "eg8199877365fc70299089371c043b96"
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir StartRxNormInferenceJobla section Référence des AWS CLI commandes.

`start-snomedct-inference-job`

L'exemple de code suivant montre comment utiliser`start-snomedct-inference-job`.

AWS CLI

Pour démarrer une tâche d'inférence SNOMED CT

L'`start-snomedct-inference-job`exemple suivant démarre une tâche d'analyse par lots par inférence SNOMED CT.

```
aws comprehendmedical start-snomedct-inference-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

Sortie :

```
{
    "JobId": "dg7289877365fc70299089371c043b96"
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir StartSnomedctInferenceJobla section Référence des AWS CLI commandes.

`stop-entities-detection-v2-job`

L'exemple de code suivant montre comment utiliser`stop-entities-detection-v2-job`.

AWS CLI

Pour arrêter une tâche de détection d'entités

L'`stop-entities-detection-v2-job`exemple suivant arrête une tâche de détection d'entités asynchrones.

```
aws comprehendmedical stop-entities-detection-v2-job \
    --job-id "ab9887877365fe70299089371c043b96"
```

Sortie :

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, consultez StopEntitiesDetectionV2Job dans AWS CLI Command Reference.

`stop-icd10-cm-inference-job`

L'exemple de code suivant montre comment utiliser`stop-icd10-cm-inference-job`.

AWS CLI

Pour arrêter une tâche d'inférence ICD-10-CM

L'`stop-icd10-cm-inference-job`exemple suivant arrête une tâche d'analyse par lots d'inférence ICD-10-CM.

```
aws comprehendmedical stop-icd10-cm-inference-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Sortie :

```
{
    "JobId": "ef7289877365fc70299089371c043b96",
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, reportez-vous à la section StopIcd10 CmInferenceJob de AWS CLI la section Référence des commandes.

`stop-phi-detection-job`

L'exemple de code suivant montre comment utiliser`stop-phi-detection-job`.

AWS CLI

Pour arrêter une tâche de détection d'informations médicales protégées (PHI)

L'`stop-phi-detection-job`exemple suivant arrête une tâche de détection asynchrone d'informations de santé protégées (PHI).

```
aws comprehendmedical stop-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Sortie :

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

Pour plus d'informations, consultez Batch APIs dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir StopPhiDetectionJobla section Référence des AWS CLI commandes.

`stop-rx-norm-inference-job`

L'exemple de code suivant montre comment utiliser`stop-rx-norm-inference-job`.

AWS CLI

Pour arrêter une tâche d' RxNorm inférence

L'`stop-rx-norm-inference-job`exemple suivant arrête une tâche d'analyse par lots d'inférence ICD-10-CM.

```
aws comprehendmedical stop-rx-norm-inference-job \
    --job-id "eg8199877365fc70299089371c043b96"
```

Sortie :

```
{
    "JobId": "eg8199877365fc70299089371c043b96",
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir StopRxNormInferenceJobla section Référence des AWS CLI commandes.

`stop-snomedct-inference-job`

L'exemple de code suivant montre comment utiliser`stop-snomedct-inference-job`.

AWS CLI

Pour arrêter une tâche d'inférence SNOMED CT

L'`stop-snomedct-inference-job`exemple suivant arrête une tâche d'analyse par lots par inférence SNOMED CT.

```
aws comprehendmedical stop-snomedct-inference-job \
    --job-id "8750034166436cdb52ffa3295a1b00a1"
```

Sortie :

```
{
    "JobId": "8750034166436cdb52ffa3295a1b00a1",
}
```

Pour plus d'informations, consultez Ontology linking batch analysis dans le manuel Amazon Comprehend Medical Developer Guide.

- Pour plus de détails sur l'API, voir StopSnomedctInferenceJobla section Référence des AWS CLI commandes.

AWS Config exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Config.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-config-rule`

L'exemple de code suivant montre comment utiliser`delete-config-rule`.

AWS CLI

Pour supprimer une règle AWS Config

La commande suivante supprime une règle AWS Config nommée `MyConfigRule` :

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- Pour plus de détails sur l'API, reportez-vous DeleteConfigRuleà la section Référence des AWS CLI commandes.

`delete-delivery-channel`

L'exemple de code suivant montre comment utiliser`delete-delivery-channel`.

AWS CLI

Pour supprimer un canal de diffusion

La commande suivante supprime le canal de diffusion par défaut :

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- Pour plus de détails sur l'API, reportez-vous DeleteDeliveryChannelà la section Référence des AWS CLI commandes.

`delete-evaluation-results`

L'exemple de code suivant montre comment utiliser`delete-evaluation-results`.

AWS CLI

Pour supprimer manuellement les résultats d'évaluation

La commande suivante supprime les résultats d'évaluation actuels pour la règle AWS gérée s3- : bucket-versioning-enabled

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- Pour plus de détails sur l'API, reportez-vous DeleteEvaluationResultsà la section Référence des AWS CLI commandes.

`deliver-config-snapshot`

L'exemple de code suivant montre comment utiliser`deliver-config-snapshot`.

AWS CLI

Pour fournir un instantané de configuration

La commande suivante fournit un instantané de configuration au compartiment Amazon S3 qui appartient au canal de diffusion par défaut :

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

Sortie :

```
{
    "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- Pour plus de détails sur l'API, reportez-vous DeliverConfigSnapshotà la section Référence des AWS CLI commandes.

`describe-compliance-by-config-rule`

L'exemple de code suivant montre comment utiliser`describe-compliance-by-config-rule`.

AWS CLI

Pour obtenir des informations de conformité pour vos règles de AWS Config

La commande suivante renvoie des informations de conformité pour chaque règle de AWS Config violée par une ou plusieurs AWS ressources :

```
aws configservice describe-compliance-by-config-rule --compliance-
types NON_COMPLIANT
```

Dans la sortie, la valeur de chaque `CappedCount` attribut indique le nombre de ressources qui ne sont pas conformes à la règle correspondante. Par exemple, le résultat suivant indique que 3 ressources ne sont pas conformes à la règle nommée`InstanceTypesAreT2micro`.

Sortie :

```
{
    "ComplianceByConfigRules": [
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "InstanceTypesAreT2micro"
        },
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 10,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
```

```
            },
            "ConfigRuleName": "RequiredTagsForVolumes"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeComplianceByConfigRuleà la section Référence des AWS CLI commandes.

`describe-compliance-by-resource`

L'exemple de code suivant montre comment utiliser`describe-compliance-by-resource`.

AWS CLI

Pour obtenir des informations de conformité pour vos AWS ressources

La commande suivante renvoie des informations de conformité pour chaque EC2 instance enregistrée par AWS Config qui enfreint une ou plusieurs règles :

```
aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
 --compliance-types NON_COMPLIANT
```

Dans le résultat, la valeur de chaque `CappedCount` attribut indique le nombre de règles violées par la ressource. Par exemple, le résultat suivant indique que l'instance `i-1a2b3c4d` enfreint 2 règles.

Sortie :

```
{
    "ComplianceByResources": [
        {
            "ResourceType": "AWS::EC2::Instance",
            "ResourceId": "i-1a2b3c4d",
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 2,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            }
        },
```

```
        {
            "ResourceType": "AWS::EC2::Instance",
            "ResourceId": "i-2a2b3c4d ",
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeComplianceByResourceà la section Référence des AWS CLI commandes.

`describe-config-rule-evaluation-status`

L'exemple de code suivant montre comment utiliser`describe-config-rule-evaluation-status`.

AWS CLI

Pour obtenir des informations sur le statut d'une règle AWS Config

La commande suivante renvoie les informations d'état d'une règle AWS Config nommée `MyConfigRule` :

```
aws configservice describe-config-rule-evaluation-status --config-rule-
names MyConfigRule
```

Sortie :

```
{
    "ConfigRulesEvaluationStatus": [
        {
            "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
            "FirstActivatedTime": 1450311703.844,
            "ConfigRuleId": "config-rule-abcdef",
            "LastSuccessfulInvocationTime": 1450314643.156,
```

```
            "ConfigRuleName": "MyConfigRule"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConfigRuleEvaluationStatusà la section Référence des AWS CLI commandes.

`describe-config-rules`

L'exemple de code suivant montre comment utiliser`describe-config-rules`.

AWS CLI

Pour obtenir les détails d'une règle AWS Config

La commande suivante renvoie les détails d'une règle AWS Config nommée `InstanceTypesAreT2micro` :

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

Sortie :

```
{
    "ConfigRules": [
        {
            "ConfigRuleState": "ACTIVE",
            "Description": "Evaluates whether EC2 instances are the t2.micro type.",
            "ConfigRuleName": "InstanceTypesAreT2micro",
            "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
            "Source": {
                "Owner": "CUSTOM_LAMBDA",
                "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
                "SourceDetails": [
                    {
                        "EventSource": "aws.config",
                        "MessageType": "ConfigurationItemChangeNotification"
                    }
                ]
            },
```

```
            "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}",
            "Scope": {
                "ComplianceResourceTypes": [
                    "AWS::EC2::Instance"
                ]
            },
            "ConfigRuleId": "config-rule-abcdef"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConfigRulesà la section Référence des AWS CLI commandes.

`describe-configuration-recorder-status`

L'exemple de code suivant montre comment utiliser`describe-configuration-recorder-status`.

AWS CLI

Pour obtenir des informations sur l'état de l'enregistreur de configuration

La commande suivante renvoie l'état de l'enregistreur de configuration par défaut :

```
aws configservice describe-configuration-recorder-status
```

Sortie :

```
{
    "ConfigurationRecordersStatus": [
        {
            "name": "default",
            "lastStatus": "SUCCESS",
            "recording": true,
            "lastStatusChangeTime": 1452193834.344,
            "lastStartTime": 1441039997.819,
            "lastStopTime": 1441039992.835
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConfigurationRecorderStatusà la section Référence des AWS CLI commandes.

`describe-configuration-recorders`

L'exemple de code suivant montre comment utiliser`describe-configuration-recorders`.

AWS CLI

Pour obtenir des informations sur l'enregistreur de configuration

La commande suivante renvoie des informations sur l'enregistreur de configuration par défaut :

```
aws configservice describe-configuration-recorders
```

Sortie :

```
{
    "ConfigurationRecorders": [
        {
            "recordingGroup": {
                "allSupported": true,
                "resourceTypes": [],
                "includeGlobalResourceTypes": true
            },
            "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-
A1B2C3D4E5F6",
            "name": "default"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConfigurationRecordersà la section Référence des AWS CLI commandes.

`describe-delivery-channel-status`

L'exemple de code suivant montre comment utiliser`describe-delivery-channel-status`.

AWS CLI

Pour obtenir des informations sur le statut du canal de distribution

La commande suivante renvoie l'état du canal de diffusion :

```
aws configservice describe-delivery-channel-status
```

Sortie :

```
{
    "DeliveryChannelsStatus": [
        {
            "configStreamDeliveryInfo": {
                "lastStatusChangeTime": 1452193834.381,
                "lastStatus": "SUCCESS"
            },
            "configHistoryDeliveryInfo": {
                "lastSuccessfulTime": 1450317838.412,
                "lastStatus": "SUCCESS",
                "lastAttemptTime": 1450317838.412
            },
            "configSnapshotDeliveryInfo": {
                "lastSuccessfulTime": 1452185597.094,
                "lastStatus": "SUCCESS",
                "lastAttemptTime": 1452185597.094
            },
            "name": "default"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDeliveryChannelStatusà la section Référence des AWS CLI commandes.

`describe-delivery-channels`

L'exemple de code suivant montre comment utiliser`describe-delivery-channels`.

AWS CLI

Pour obtenir des informations sur le canal de livraison

La commande suivante renvoie des informations sur le canal de diffusion :

```
aws configservice describe-delivery-channels
```

Sortie :

```
{
    "DeliveryChannels": [
        {
            "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
            "name": "default",
            "s3BucketName": "config-bucket-123456789012"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDeliveryChannelsà la section Référence des AWS CLI commandes.

`get-compliance-details-by-config-rule`

L'exemple de code suivant montre comment utiliser`get-compliance-details-by-config-rule`.

AWS CLI

Pour obtenir les résultats de l'évaluation d'une règle AWS Config

La commande suivante renvoie les résultats de l'évaluation pour toutes les ressources qui ne sont pas conformes à une règle de AWS configuration nommée `InstanceTypesAreT2micro` :

```
aws configservice get-compliance-details-by-config-rule --config-rule-
name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```

Sortie :

```
{
    "EvaluationResults": [
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-1a2b3c4d",
```

```
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314645.261,
            "ConfigRuleInvokedTime": 1450314642.948,
            "ComplianceType": "NON_COMPLIANT"
        },
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-2a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314645.18,
            "ConfigRuleInvokedTime": 1450314642.902,
            "ComplianceType": "NON_COMPLIANT"
        },
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-3a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314643.346,
            "ConfigRuleInvokedTime": 1450314643.124,
            "ComplianceType": "NON_COMPLIANT"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetComplianceDetailsByConfigRuleà la section Référence des AWS CLI commandes.

`get-compliance-details-by-resource`

L'exemple de code suivant montre comment utiliser`get-compliance-details-by-resource`.

AWS CLI

Pour obtenir les résultats de l'évaluation d'une AWS ressource

La commande suivante renvoie les résultats de l'évaluation pour chaque règle à laquelle l' EC2 instance `i-1a2b3c4d` n'est pas conforme :

```
aws configservice get-compliance-details-by-resource --resource-
type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

Sortie :

```
{
    "EvaluationResults": [
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-1a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314643.288,
            "ConfigRuleInvokedTime": 1450314643.034,
            "ComplianceType": "NON_COMPLIANT"
        },
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-1a2b3c4d",
                    "ConfigRuleName": "RequiredTagForEC2Instances"
                }
            },
            "ResultRecordedTime": 1450314645.261,
            "ConfigRuleInvokedTime": 1450314642.948,
            "ComplianceType": "NON_COMPLIANT"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetComplianceDetailsByResourceà la section Référence des AWS CLI commandes.

`get-compliance-summary-by-config-rule`

L'exemple de code suivant montre comment utiliser`get-compliance-summary-by-config-rule`.

AWS CLI

Pour obtenir le résumé de conformité de vos règles de AWS Config

La commande suivante renvoie le nombre de règles conformes et le nombre de règles non conformes :

```
aws configservice get-compliance-summary-by-config-rule
```

Dans la sortie, la valeur de chaque `CappedCount` attribut indique le nombre de règles conformes ou non conformes.

Sortie :

```
{
    "ComplianceSummary": {
        "NonCompliantResourceCount": {
            "CappedCount": 3,
            "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1452204131.493,
        "CompliantResourceCount": {
            "CappedCount": 2,
            "CapExceeded": false
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetComplianceSummaryByConfigRuleà la section Référence des AWS CLI commandes.

`get-compliance-summary-by-resource-type`

L'exemple de code suivant montre comment utiliser`get-compliance-summary-by-resource-type`.

AWS CLI

Pour obtenir le résumé de conformité pour tous les types de ressources

La commande suivante renvoie le nombre de AWS ressources non conformes et le nombre de ressources conformes :

```
aws configservice get-compliance-summary-by-resource-type
```

Dans la sortie, la valeur de chaque `CappedCount` attribut indique le nombre de ressources conformes ou non conformes.

Sortie :

```
{
    "ComplianceSummariesByResourceType": [
        {
            "ComplianceSummary": {
                "NonCompliantResourceCount": {
                    "CappedCount": 16,
                    "CapExceeded": false
                },
                "ComplianceSummaryTimestamp": 1453237464.543,
                "CompliantResourceCount": {
                    "CappedCount": 10,
                    "CapExceeded": false
                }
            }
        }
    ]
}
```

Pour obtenir le résumé de conformité pour un type de ressource spécifique

La commande suivante renvoie le nombre d' EC2 instances non conformes et le nombre d'instances conformes :

```
aws configservice get-compliance-summary-by-resource-type --resource-
types AWS::EC2::Instance
```

Dans la sortie, la valeur de chaque `CappedCount` attribut indique le nombre de ressources conformes ou non conformes.

Sortie :

```
{
    "ComplianceSummariesByResourceType": [
        {
            "ResourceType": "AWS::EC2::Instance",
            "ComplianceSummary": {
                "NonCompliantResourceCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceSummaryTimestamp": 1452204923.518,
                "CompliantResourceCount": {
                    "CappedCount": 7,
                    "CapExceeded": false
                }
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetComplianceSummaryByResourceTypeà la section Référence des AWS CLI commandes.

`get-resource-config-history`

L'exemple de code suivant montre comment utiliser`get-resource-config-history`.

AWS CLI

Pour obtenir l'historique de configuration d'une AWS ressource

La commande suivante renvoie une liste d'éléments de configuration pour une EC2 instance avec un ID de `i-1a2b3c4d` :

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --
resource-id i-1a2b3c4d
```

- Pour plus de détails sur l'API, reportez-vous GetResourceConfigHistoryà la section Référence des AWS CLI commandes.

`get-status`

L'exemple de code suivant montre comment utiliser`get-status`.

AWS CLI

Pour obtenir le statut de AWS Config

La commande suivante renvoie l'état du canal de diffusion et de l'enregistreur de configuration :

```
aws configservice get-status
```

Sortie :

```
Configuration Recorders:

name: default
recorder: ON
last status: SUCCESS

Delivery Channels:

name: default
last stream delivery status: SUCCESS
last history delivery status: SUCCESS
last snapshot delivery status: SUCCESS
```

- Pour plus de détails sur l'API, reportez-vous GetStatusà la section Référence des AWS CLI commandes.

`list-discovered-resources`

L'exemple de code suivant montre comment utiliser`list-discovered-resources`.

AWS CLI

Pour répertorier les ressources découvertes par AWS Config

La commande suivante répertorie les EC2 instances découvertes par AWS Config :

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

Sortie :

```
{
    "resourceIdentifiers": [
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-1a2b3c4d"
        },
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-2a2b3c4d"
        },
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-3a2b3c4d"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDiscoveredResourcesà la section Référence des AWS CLI commandes.

`put-config-rule`

L'exemple de code suivant montre comment utiliser`put-config-rule`.

AWS CLI

Pour ajouter une règle Config AWS gérée

La commande suivante fournit du code JSON pour ajouter une règle de configuration AWS gérée :

```
aws configservice put-config-rule --config-rule file://
RequiredTagsForEC2Instances.json
```

`RequiredTagsForEC2Instances.json`est un fichier JSON qui contient la configuration des règles :

```
{
  "ConfigRuleName": "RequiredTagsForEC2Instances",
  "Description": "Checks whether the CostCenter and Owner tags are applied to EC2
 instances.",
  "Scope": {
    "ComplianceResourceTypes": [
      "AWS::EC2::Instance"
    ]
  },
  "Source": {
    "Owner": "AWS",
    "SourceIdentifier": "REQUIRED_TAGS"
  },
  "InputParameters": "{\"tag1Key\":\"CostCenter\",\"tag2Key\":\"Owner\"}"
}
```

Pour l'`ComplianceResourceTypes`attribut, ce code JSON limite la portée aux ressources de ce `AWS::EC2::Instance` type, de sorte que AWS Config évaluera uniquement les EC2 instances par rapport à la règle. Comme la règle est une règle gérée, l'`Owner`attribut est défini sur`AWS`, et l'`SourceIdentifier`attribut est défini sur l'identifiant de règle,`REQUIRED_TAGS`. Pour l'`InputParameters`attribut, les clés de balise requises par la règle, `CostCenter` et`Owner`, sont spécifiées.

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier la configuration de la règle, exécutez la describe-config-rules commande et spécifiez le nom de la règle.

Pour ajouter une règle Config gérée par le client

La commande suivante fournit du code JSON pour ajouter une règle de configuration gérée par le client :

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

`InstanceTypesAreT2micro.json`est un fichier JSON qui contient la configuration des règles :

```
{
  "ConfigRuleName": "InstanceTypesAreT2micro",
  "Description": "Evaluates whether EC2 instances are the t2.micro type.",
  "Scope": {
    "ComplianceResourceTypes": [
      "AWS::EC2::Instance"
    ]
  },
  "Source": {
    "Owner": "CUSTOM_LAMBDA",
    "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
    "SourceDetails": [
      {
        "EventSource": "aws.config",
        "MessageType": "ConfigurationItemChangeNotification"
      }
    ]
  },
  "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}"
}
```

Pour l'`ComplianceResourceTypes`attribut, ce code JSON limite la portée aux ressources de ce `AWS::EC2::Instance` type, de sorte que AWS Config évaluera uniquement les EC2 instances par rapport à la règle. Cette règle étant une règle gérée par le client, l'`Owner`attribut est défini sur`CUSTOM_LAMBDA`, et l'`SourceIdentifier`attribut est défini sur l'ARN de la fonction AWS Lambda. L'`SourceDetails`objet est obligatoire. Les paramètres spécifiés pour l'`InputParameters`attribut sont transmis à la fonction AWS Lambda lorsque AWS Config l'invoque pour évaluer les ressources par rapport à la règle.

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier la configuration de la règle, exécutez la describe-config-rules commande et spécifiez le nom de la règle.

- Pour plus de détails sur l'API, reportez-vous PutConfigRuleà la section Référence des AWS CLI commandes.

`put-configuration-recorder`

L'exemple de code suivant montre comment utiliser`put-configuration-recorder`.

AWS CLI

Exemple 1 : pour enregistrer toutes les ressources prises en charge

La commande suivante crée un enregistreur de configuration qui suit les modifications apportées à tous les types de ressources pris en charge, y compris les types de ressources globaux :

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/config-role \
    --recording-group allSupported=true,includeGlobalResourceTypes=true
```

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier les paramètres de votre enregistreur de configuration, exécutez la describe-configuration-recorders commande.

Exemple 2 : pour enregistrer des types de ressources spécifiques

La commande suivante crée un enregistreur de configuration qui suit les modifications apportées uniquement aux types de ressources spécifiés dans le fichier JSON pour l'option --recording-group :

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/config-role \
    --recording-group file://recordingGroup.json
```

RecordingGroup.json est un fichier JSON qui spécifie les types de ressources que Config AWS enregistrera :

```
{
    "allSupported": false,
    "includeGlobalResourceTypes": false,
    "resourceTypes": [
        "AWS::EC2::EIP",
        "AWS::EC2::Instance",
        "AWS::EC2::NetworkAcl",
        "AWS::EC2::SecurityGroup",
        "AWS::CloudTrail::Trail",
        "AWS::EC2::Volume",
        "AWS::EC2::VPC",
        "AWS::IAM::User",
        "AWS::IAM::Policy"
```

```
    ]
}
```

Avant de pouvoir spécifier des types de ressources pour la clé ResourceTypes, vous devez définir les options AllSupported includeGlobalResource et Types sur false ou les omettre.

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier les paramètres de votre enregistreur de configuration, exécutez la describe-configuration-recorders commande.

Exemple 3 : Pour sélectionner toutes les ressources prises en charge, à l'exception de certains types de ressources

La commande suivante crée un enregistreur de configuration qui suit les modifications apportées à tous les types de ressources pris en charge actuels et futurs, à l'exception des types de ressources spécifiés dans le fichier JSON pour l'option --recording-group :

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/
config-role \
    --recording-group file://recordingGroup.json
```

RecordingGroup.json est un fichier JSON qui spécifie les types de ressources que Config AWS enregistrera :

```
{
    "allSupported": false,
    "exclusionByResourceTypes": {
        "resourceTypes": [
        "AWS::Redshift::ClusterSnapshot",
        "AWS::RDS::DBClusterSnapshot",
        "AWS::CloudFront::StreamingDistribution"
    ]
    },
        "includeGlobalResourceTypes": false,
        "recordingStrategy": {
        "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"
    },
}
```

Avant de pouvoir spécifier les types de ressources à exclure de l'enregistrement : 1) Vous devez définir les options AllSupported et includeGlobalResource Types sur false ou les

omettre, et 2) Vous devez définir le champ UseOnly sur EXCLUSION_BY_RESOURCE_TYPES RecordingStrategy .

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier les paramètres de votre enregistreur de configuration, exécutez la describe-configuration-recorders commande.

- Pour plus de détails sur l'API, reportez-vous PutConfigurationRecorderà la section Référence des AWS CLI commandes.

`put-delivery-channel`

L'exemple de code suivant montre comment utiliser`put-delivery-channel`.

AWS CLI

Pour créer un canal de diffusion

La commande suivante fournit les paramètres du canal de diffusion sous forme de code JSON :

```
aws configservice put-delivery-channel --delivery-channel file://
deliveryChannel.json
```

Le `deliveryChannel.json` fichier spécifie les attributs du canal de diffusion :

```
{
    "name": "default",
    "s3BucketName": "config-bucket-123456789012",
    "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
    "configSnapshotDeliveryProperties": {
        "deliveryFrequency": "Twelve_Hours"
    }
}
```

Cet exemple définit les attributs suivants :

`name`- Le nom du canal de diffusion. Par défaut, AWS Config attribue le nom `default` à un nouveau canal de distribution. Vous ne pouvez pas mettre à jour le nom du canal de distribution avec la commande. `put-delivery-channel` Pour connaître les étapes à suivre pour modifier le nom, voir Renommer le canal de diffusion. `s3BucketName` - Le nom du compartiment Amazon S3 auquel AWS Config fournit des instantanés de configuration et des fichiers d'historique

de configuration. Si vous spécifiez un compartiment appartenant à un autre AWS compte, ce compartiment doit disposer de politiques accordant des autorisations d'accès à Config. AWS Pour de plus d'informations, consultez Autorisations pour le compartiment Amazon S3.

`snsTopicARN`- Le nom de ressource Amazon (ARN) de la rubrique Amazon SNS à laquelle AWS Config envoie des notifications concernant les modifications de configuration. Si vous choisissez un sujet depuis un autre compte, le sujet doit comporter des politiques accordant des autorisations d'accès à Config. AWS Pour plus d'informations, consultez la rubrique Autorisations pour Amazon SNS.

`configSnapshotDeliveryProperties`- Contient l'`deliveryFrequency`attribut, qui définit la fréquence à laquelle AWS Config fournit des instantanés de configuration et la fréquence à laquelle il invoque des évaluations pour les règles de configuration périodiques.

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier les paramètres de votre canal de diffusion, exécutez la describe-delivery-channels commande.

- Pour plus de détails sur l'API, reportez-vous PutDeliveryChannelà la section Référence des AWS CLI commandes.

`start-config-rules-evaluation`

L'exemple de code suivant montre comment utiliser`start-config-rules-evaluation`.

AWS CLI

Pour exécuter une évaluation à la demande des règles AWS Config

La commande suivante lance une évaluation pour deux règles AWS gérées :

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-versioning-enabled cloudtrail-enabled
```

- Pour plus de détails sur l'API, reportez-vous StartConfigRulesEvaluationà la section Référence des AWS CLI commandes.

`start-configuration-recorder`

L'exemple de code suivant montre comment utiliser`start-configuration-recorder`.

AWS CLI

Pour démarrer l'enregistreur de configuration

La commande suivante démarre l'enregistreur de configuration par défaut :

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier que AWS Config enregistre vos ressources, exécutez la commande get-status.

- Pour plus de détails sur l'API, reportez-vous StartConfigurationRecorderà la section Référence des AWS CLI commandes.

`stop-configuration-recorder`

L'exemple de code suivant montre comment utiliser`stop-configuration-recorder`.

AWS CLI

Pour arrêter l'enregistreur de configuration

La commande suivante arrête l'enregistreur de configuration par défaut :

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

Si la commande aboutit, AWS Config ne renvoie aucune sortie. Pour vérifier que AWS Config n'enregistre pas vos ressources, exécutez la commande get-status.

- Pour plus de détails sur l'API, reportez-vous StopConfigurationRecorderà la section Référence des AWS CLI commandes.

`subscribe`

L'exemple de code suivant montre comment utiliser`subscribe`.

AWS CLI

Pour vous abonner à AWS Config

La commande suivante crée le canal de diffusion et l'enregistreur de configuration par défaut. La commande spécifie également le compartiment Amazon S3 et la rubrique Amazon SNS auxquels AWS Config fournira les informations de configuration :

```
aws configservice subscribe --s3-bucket config-bucket-123456789012
 --sns-topic arn:aws:sns:us-east-1:123456789012:config-topic --iam-
role arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6
```

Sortie :

```
Using existing S3 bucket: config-bucket-123456789012
Using existing SNS topic: arn:aws:sns:us-east-1:123456789012:config-topic
Subscribe succeeded:

Configuration Recorders: [
    {
        "recordingGroup": {
            "allSupported": true,
            "resourceTypes": [],
            "includeGlobalResourceTypes": false
        },
        "roleARN": "arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6",
        "name": "default"
    }
]

Delivery Channels: [
    {
        "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
        "name": "default",
        "s3BucketName": "config-bucket-123456789012"
    }
]
```

- Pour plus d'informations sur l'API, consultez Subscribe dans la Référence des commandes AWS CLI .

Exemples d'Amazon Connect utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Connect.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un utilisateur

L'`create-user`exemple suivant ajoute un utilisateur avec les attributs spécifiés à l'instance Amazon Connect spécifiée.

```
aws connect create-user \
    --username Mary \
    --password Pass@Word1 \
    --identity-info FirstName=Mary,LastName=Major \
    --phone-
config PhoneType=DESK_PHONE,AutoAccept=true,AfterContactWorkTimeLimit=60,DeskPhoneNumber=
+15555551212 \
    --security-profile-id 12345678-1111-2222-aaaa-a1b2c3d4f5g7 \
    --routing-profile-id 87654321-9999-3434-abcd-x1y2z3a1b2c3 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "UserId": "87654321-2222-1234-1234-111234567891",
    "UserArn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111/agent/87654321-2222-1234-1234-111234567891"
}
```

Pour plus d'informations, consultez la section Ajouter des utilisateurs dans le manuel Amazon Connect Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

L'`delete-user`exemple suivant supprime l'utilisateur spécifié de l'instance Amazon Connect spécifiée.

```
aws connect delete-user \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --user-id 87654321-2222-1234-1234-111234567891
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gérer les utilisateurs dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`describe-user-hierarchy-group`

L'exemple de code suivant montre comment utiliser`describe-user-hierarchy-group`.

AWS CLI

Pour afficher les détails d'un groupe hiérarchique

L'`describe-user-hierarchy-group`exemple suivant affiche les détails du groupe hiérarchique Amazon Connect spécifié.

```
aws connect describe-user-hierarchy-group \
    --hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \
```

```
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
     "HierarchyGroup": {
         "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
         "Arn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111/agent-group/12345678-1111-2222-800e-a2b3c4d5f6g7",
         "Name": "Example Corporation",
         "LevelId": "1",
         "HierarchyPath": {
             "LevelOne": {
                 "Id": "abcdefgh-3333-4444-8af3-201123456789",
                 "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group/
abcdefgh-3333-4444-8af3-201123456789",
                 "Name": "Example Corporation"
             }
         }
     }
 }
```

Pour plus d'informations, consultez Configurer les hiérarchies d'agents dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous DescribeUserHierarchyGroupà la section Référence des AWS CLI commandes.

`describe-user-hierarchy-structure`

L'exemple de code suivant montre comment utiliser`describe-user-hierarchy-structure`.

AWS CLI

Pour afficher les détails d'une structure hiérarchique

L'`describe-user-hierarchy-structure`exemple suivant affiche les détails de la structure hiérarchique de l'instance Amazon Connect spécifiée.

```
aws connect describe-user-hierarchy-group \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
     "HierarchyStructure": {
         "LevelOne": {
             "Id": "12345678-1111-2222-800e-aaabbb555gg",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/1",
             "Name": "Corporation"
         },
         "LevelTwo": {
             "Id": "87654321-2222-3333-ac99-123456789102",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/2",
             "Name": "Services Division"
         },
         "LevelThree": {
             "Id": "abcdefgh-3333-4444-8af3-201123456789",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/3",
             "Name": "EU Site"
         }
     }
 }
```

Pour plus d'informations, consultez Configurer les hiérarchies d'agents dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous DescribeUserHierarchyStructureà la section Référence des AWS CLI commandes.

`describe-user`

L'exemple de code suivant montre comment utiliser`describe-user`.

AWS CLI

Pour afficher les informations relatives à un utilisateur

L'`describe-user`exemple suivant affiche les détails de l'utilisateur Amazon Connect spécifié.

```
aws connect describe-user \
    --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60
```

```
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Sortie :

```
{
    "User": {
        "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
        "Username": "Jane",
        "IdentityInfo": {
            "FirstName": "Jane",
            "LastName": "Doe",
            "Email": "example.com"
        },
        "PhoneConfig": {
            "PhoneType": "SOFT_PHONE",
            "AutoAccept": false,
            "AfterContactWorkTimeLimit": 0,
            "DeskPhoneNumber": ""
        },
        "DirectoryUserId": "8b444cf6-b368-4f29-ba18-07af27405658",
        "SecurityProfileIds": [
            "b6f85a42-1dc5-443b-b621-de0abf70c9cf"
        ],
        "RoutingProfileId": "0be36ee9-2b5f-4ef4-bcf7-87738e5be0e5",
        "Tags": {}
    }
}
```

Pour plus d'informations, consultez la section Gérer les utilisateurs dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous DescribeUserà la section Référence des AWS CLI commandes.

`get-contact-attributes`

L'exemple de code suivant montre comment utiliser`get-contact-attributes`.

AWS CLI

Pour récupérer les attributs d'un contact

L'`get-contact-attributes`exemple suivant récupère les attributs définis pour le contact Amazon Connect spécifié.

```
aws connect get-contact-attributes \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7
```

Sortie :

```
{
    "Attributes": {
        "greetingPlayed": "true"
    }
}
```

Pour plus d'informations, consultez la section Utiliser les attributs de contact Amazon Connect dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous GetContactAttributesà la section Référence des AWS CLI commandes.

`list-contact-flows`

L'exemple de code suivant montre comment utiliser`list-contact-flows`.

AWS CLI

Pour répertorier les flux de contacts dans une instance

L'`list-contact-flows`exemple suivant répertorie les flux de contacts dans l'instance Amazon Connect spécifiée.

```
aws connect list-contact-flows \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
     "ContactFlowSummaryList": [
```

```
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "Name": "Default queue transfer",
            "ContactFlowType": "QUEUE_TRANSFER"
        },
        {
            "Id": "87654321-2222-3333-ac99-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/87654321-2222-3333-
ac99-123456789102",
            "Name": "Default agent hold",
            "ContactFlowType": "AGENT_HOLD"
        },
        {
            "Id": "abcdefgh-3333-4444-8af3-201123456789",
                "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/
abcdefgh-3333-4444-8af3-201123456789",
            "Name": "Default customer hold",
            "ContactFlowType": "CUSTOMER_HOLD"
        },
    ]
 }
```

Pour plus d'informations, consultez la section Créer des flux de contacts Amazon Connect dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListContactFlowsà la section Référence des AWS CLI commandes.

`list-hours-of-operations`

L'exemple de code suivant montre comment utiliser`list-hours-of-operations`.

AWS CLI

Pour répertorier les heures d'ouverture d'une instance

L'`list-hours-of-operations`exemple suivant répertorie les heures d'ouverture de l'instance Amazon Connect spécifiée.

```
aws connect list-hours-of-operations \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Sortie :

```
{
    "HoursOfOperationSummaryList": [
        {
            "Id": "d69f1f84-7457-4924-8fbe-e64875546259",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/operating-hours/d69f1f84-7457-4924-8fbe-e64875546259",
            "Name": "Basic Hours"
        }
    ]
}
```

Pour plus d'informations, consultez la section Définir les heures d'ouverture d'une file d'attente dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListHoursOfOperationsà la section Référence des AWS CLI commandes.

`list-phone-numbers`

L'exemple de code suivant montre comment utiliser`list-phone-numbers`.

AWS CLI

Pour répertorier les numéros de téléphone d'une instance

L'`list-phone-numbers`exemple suivant répertorie les numéros de téléphone de l'instance Amazon Connect spécifiée.

```
aws connect list-phone-numbers \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "PhoneNumberSummaryList": [
```

```
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/xyz80zxy-xyz1-80zx-
zx80-11111EXAMPLE",
            "PhoneNumber": "+17065551212",
            "PhoneNumberType": "DID",
            "PhoneNumberCountryCode": "US"
        },
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/ccc0ccc-xyz1-80zx-
zx80-22222EXAMPLE",
            "PhoneNumber": "+18555551212",
            "PhoneNumberType": "TOLL_FREE",
            "PhoneNumberCountryCode": "US"
        }
    ]
}
```

Pour plus d'informations, consultez Configurer les numéros de téléphone de votre centre d'appels dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListPhoneNumbersà la section Référence des AWS CLI commandes.

`list-queues`

L'exemple de code suivant montre comment utiliser`list-queues`.

AWS CLI

Pour répertorier les files d'attente dans une instance

L'`list-queues`exemple suivant répertorie les files d'attente dans l'instance Amazon Connect spécifiée.

```
aws connect list-queues \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "QueueSummaryList": [
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "QueueType": "AGENT"
        },
        {
            "Id": "87654321-2222-3333-ac99-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/87654321-2222-3333-
ac99-123456789102",
            "QueueType": "AGENT"
        },
        {
            "Id": "abcdefgh-3333-4444-8af3-201123456789",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/
abcdefgh-3333-4444-8af3-201123456789",
            "QueueType": "AGENT"
        },
        {
            "Id": "hgfedcba-4444-5555-a31f-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/hgfedcba-4444-5555-a31f-123456789102",
            "Name": "BasicQueue",
            "QueueType": "STANDARD"
        },
    ]
}
```

Pour plus d'informations, consultez la section Créer une file d'attente dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListQueuesà la section Référence des AWS CLI commandes.

`list-routing-profiles`

L'exemple de code suivant montre comment utiliser`list-routing-profiles`.

AWS CLI

Pour répertorier les profils de routage dans une instance

L'`list-routing-profiles`exemple suivant répertorie les profils de routage dans l'instance Amazon Connect spécifiée.

```
aws connect list-routing-profiles \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RoutingProfileSummaryList": [
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/routing-profile/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "Name": "Basic Routing Profile"
        },
    ]
}
```

Pour plus d'informations, consultez la section Créer un profil de routage dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListRoutingProfilesà la section Référence des AWS CLI commandes.

`list-security-profiles`

L'exemple de code suivant montre comment utiliser`list-security-profiles`.

AWS CLI

Pour répertorier les profils de sécurité d'une instance

L'`list-security-profiles`exemple suivant répertorie les profils de sécurité de l'instance Amazon Connect spécifiée.

```
aws connect list-security-profiles \
```

```
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "SecurityProfileSummaryList": [
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "Name": "CallCenterManager"
        },
        {
            "Id": "87654321-2222-3333-ac99-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/87654321-2222-3333-
ac99-123456789102",
            "Name": "QualityAnalyst"
        },
        {
            "Id": "abcdefgh-3333-4444-8af3-201123456789",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/
abcdefgh-3333-4444-8af3-201123456789",
            "Name": "Agent"
        },
        {
            "Id": "12345678-1111-2222-800e-x2y3c4d5fzzzz",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
x2y3c4d5fzzzz",
            "Name": "Admin"
        }
    ]
}
```

Pour plus d'informations, consultez la section Attribuer des autorisations : profils de sécurité dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListSecurityProfilesà la section Référence des AWS CLI commandes.

`list-user-hierarchy-groups`

L'exemple de code suivant montre comment utiliser`list-user-hierarchy-groups`.

AWS CLI

Pour répertorier les groupes hiérarchiques d'utilisateurs dans une instance

L'`list-user-hierarchy-groups`exemple suivant répertorie les groupes hiérarchiques d'utilisateurs dans l'instance Amazon Connect spécifiée.

```
aws connect list-user-hierarchy-groups \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Sortie :

```
{
    "UserHierarchyGroupSummaryList": [
        {
            "Id": "0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent-group/0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
            "Name": "Example Corporation"
        },
    ]
}
```

Pour plus d'informations, consultez Configurer les hiérarchies d'agents dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous ListUserHierarchyGroupsà la section Référence des AWS CLI commandes.

`list-users`

L'exemple de code suivant montre comment utiliser`list-users`.

AWS CLI

Pour répertorier les groupes hiérarchiques d'utilisateurs dans une instance

L'`list-users`exemple suivant répertorie les utilisateurs de l'instance Amazon Connect spécifiée.

```
aws connect list-users \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Sortie :

```
{
    "UserSummaryList": [
        {
            "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
            "Username": "Jane"
        },
        {
            "Id": "46f0c67c-3fc7-4806-ac99-403798788c14",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/46f0c67c-3fc7-4806-ac99-403798788c14",
            "Username": "Paulo"
        },
        {
            "Id": "55a83578-95e1-4710-8af3-2b7afe310e48",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/55a83578-95e1-4710-8af3-2b7afe310e48",
            "Username": "JohnD"
        },
        {
            "Id": "703e27b5-c9f0-4f1f-a239-64ccbb160125",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/703e27b5-c9f0-4f1f-a239-64ccbb160125",
            "Username": "JohnS"
        }
    ]
}
```

Pour plus d'informations, consultez la section Ajouter des utilisateurs dans le manuel Amazon Connect Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`update-contact-attributes`

L'exemple de code suivant montre comment utiliser`update-contact-attributes`.

AWS CLI

Pour mettre à jour l'attribut d'un contact

L'`update-contact-attributes`exemple suivant met à jour l'`greetingPlayed`attribut pour l'utilisateur Amazon Connect spécifié.

```
aws connect update-contact-attributes \
    --initial-contact-id 11111111-2222-3333-4444-12345678910 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --attributes greetingPlayed=false
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les attributs de contact Amazon Connect dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateContactAttributesà la section Référence des AWS CLI commandes.

`update-user-hierarchy`

L'exemple de code suivant montre comment utiliser`update-user-hierarchy`.

AWS CLI

Pour mettre à jour la hiérarchie d'un utilisateur

L'`update-user-hierarchy`exemple suivant met à jour la hiérarchie des agents pour l'utilisateur Amazon Connect spécifié.

```
aws connect update-user-hierarchy \
    --hierarchy-group-id 12345678-a1b2-c3d4-e5f6-123456789abc \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configurer les paramètres de l'agent dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateUserHierarchyà la section Référence des AWS CLI commandes.

`update-user-identity-info`

L'exemple de code suivant montre comment utiliser`update-user-identity-info`.

AWS CLI

Pour mettre à jour les informations d'identité d'un utilisateur

L'`update-user-identity-info`exemple suivant met à jour les informations d'identité de l'utilisateur Amazon Connect spécifié.

```
aws connect update-user-identity-info \
    --identity-info FirstName=Mary,LastName=Major,Email=marym@example.com \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configurer les paramètres de l'agent dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateUserIdentityInfoà la section Référence des AWS CLI commandes.

`update-user-phone-config`

L'exemple de code suivant montre comment utiliser`update-user-phone-config`.

AWS CLI

Pour mettre à jour la configuration du téléphone d'un utilisateur

L'`update-user-phone-config`exemple suivant met à jour la configuration du téléphone pour l'utilisateur spécifié.

```
aws connect update-user-phone-config \
```

```
    --phone-
config PhoneType=SOFT_PHONE,AutoAccept=false,AfterContactWorkTimeLimit=60,DeskPhoneNumber=
+18005551212 \
    --user-id 12345678-4444-3333-2222-111122223333 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configurer les paramètres de l'agent dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateUserPhoneConfigà la section Référence des AWS CLI commandes.

`update-user-routing-profile`

L'exemple de code suivant montre comment utiliser`update-user-routing-profile`.

AWS CLI

Pour mettre à jour le profil de routage d'un utilisateur

L'`update-user-routing-profile`exemple suivant met à jour le profil de routage pour l'utilisateur Amazon Connect spécifié.

```
aws connect update-user-routing-profile \
    --routing-profile-id 12345678-1111-3333-2222-4444EXAMPLE \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configurer les paramètres de l'agent dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateUserRoutingProfileà la section Référence des AWS CLI commandes.

`update-user-security-profiles`

L'exemple de code suivant montre comment utiliser`update-user-security-profiles`.

AWS CLI

Pour mettre à jour les profils de sécurité d'un utilisateur

L'`update-user-security-profiles`exemple suivant met à jour le profil de sécurité pour l'utilisateur Amazon Connect spécifié.

```
aws connect update-user-security-profiles \
    --security-profile-ids 12345678-1234-1234-1234-1234567892111 \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Attribuer des autorisations : profils de sécurité dans le guide de l'administrateur Amazon Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateUserSecurityProfilesà la section Référence des AWS CLI commandes.

AWS Cost and Usage Report exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Cost and Usage Report.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-report-definition`

L'exemple de code suivant montre comment utiliser`delete-report-definition`.

AWS CLI

Pour supprimer un rapport sur les AWS coûts et l'utilisation

Cet exemple supprime un rapport sur les AWS coûts et l'utilisation.

Commande :

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- Pour plus de détails sur l'API, reportez-vous DeleteReportDefinitionà la section Référence des AWS CLI commandes.

`describe-report-definitions`

L'exemple de code suivant montre comment utiliser`describe-report-definitions`.

AWS CLI

Pour récupérer la liste des rapports sur les AWS coûts et l'utilisation

Cet exemple décrit une liste de rapports sur les AWS coûts et l'utilisation détenus par un compte.

Commande :

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

Sortie :

```
  {
"ReportDefinitions": [
  {
      "ReportName": "ExampleReport",
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
          "REDSHIFT",
```

```
            "QUICKSIGHT"
        ],
        "AdditionalSchemaElements": [
            "RESOURCES"
        ]
    }
]
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeReportDefinitionsà la section Référence des AWS CLI commandes.

`put-report-definition`

L'exemple de code suivant montre comment utiliser`put-report-definition`.

AWS CLI

Pour créer un rapport sur les AWS coûts et l'utilisation

L'`put-report-definition`exemple suivant crée un rapport quotidien sur les AWS coûts et l'utilisation que vous pouvez télécharger sur Amazon Redshift ou Amazon. QuickSight

```
aws cur put-report-definition --report-definition file://report-definition.json
```

Contenu de `report-definition.json` :

```
{
    "ReportName": "ExampleReport",
    "TimeUnit": "DAILY",
    "Format": "textORcsv",
    "Compression": "ZIP",
    "AdditionalSchemaElements": [
        "RESOURCES"
    ],
    "S3Bucket": "example-s3-bucket",
    "S3Prefix": "exampleprefix",
    "S3Region": "us-east-1",
    "AdditionalArtifacts": [
        "REDSHIFT",
        "QUICKSIGHT"
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutReportDefinitionà la section Référence des AWS CLI commandes.

Exemples du service Cost Explorer utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du service AWS Command Line Interface with Cost Explorer.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-cost-and-usage`

L'exemple de code suivant montre comment utiliser`get-cost-and-usage`.

AWS CLI

Pour récupérer l'utilisation S3 d'un compte pour le mois de septembre 2017

L'`get-cost-and-usage`exemple suivant récupère l'utilisation S3 d'un compte pour le mois de septembre 2017.

```
aws ce get-cost-and-usage \
    --time-period Start=2017-09-01,End=2017-10-01 \
    --granularity MONTHLY \
    --metrics "BlendedCost" "UnblendedCost" "UsageQuantity" \
    --group-by Type=DIMENSION,Key=SERVICE Type=TAG,Key=Environment \
```

```
    --filter file://filters.json
```

Contenu de `filters.json` :

```
{
    "Dimensions": {
        "Key": "SERVICE",
        "Values": [
            "Amazon Simple Storage Service"
        ]
    }
}
```

Sortie :

```
{
    "GroupDefinitions": [
        {
            "Type": "DIMENSION",
            "Key": "SERVICE"
        },
        {
            "Type": "TAG",
            "Key": "Environment"
        }
    ],
    "ResultsByTime": [
        {
            "Estimated": false,
            "TimePeriod": {
                "Start": "2017-09-01",
                "End": "2017-10-01"
            },
            "Total": {},
            "Groups": [
                {
                    "Keys": [
                        "Amazon Simple Storage Service",
                        "Environment$"
                    ],
                    "Metrics": {
                        "BlendedCost": {
                            "Amount": "40.3527508453",
```

```
                            "Unit": "USD"
                        },
                        "UnblendedCost": {
                            "Amount": "40.3543773134",
                            "Unit": "USD"
                        },
                        "UsageQuantity": {
                            "Amount": "9312771.098461578",
                            "Unit": "N/A"
                        }
                    }
                },
                {
                    "Keys": [
                        "Amazon Simple Storage Service",
                        "Environment$Dev"
                    ],
                    "Metrics": {
                        "BlendedCost": {
                            "Amount": "0.2682364644",
                            "Unit": "USD"
                        },
                        "UnblendedCost": {
                            "Amount": "0.2682364644",
                            "Unit": "USD"
                        },
                        "UsageQuantity": {
                            "Amount": "22403.4395271182",
                            "Unit": "N/A"
                        }
                    }
                }
            ]
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous GetCostAndUsageà la section Référence des AWS CLI commandes.

`get-dimension-values`

L'exemple de code suivant montre comment utiliser`get-dimension-values`.

AWS CLI

Pour récupérer les balises de la dimension SERVICE, avec la valeur « Elastic »

Cet exemple extrait les balises de la dimension SERVICE, avec une valeur « Elastic » pour la période allant du 1er janvier 2017 au 18 mai 2017.

Commande :

```
aws ce get-dimension-values --search-string Elastic --time-
period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

Sortie :

```
{
   "TotalSize": 6,
   "DimensionValues": [
       {
           "Attributes": {},
           "Value": "Amazon ElastiCache"
       },
       {
           "Attributes": {},
           "Value": "EC2 - Other"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Compute Cloud - Compute"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Load Balancing"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic MapReduce"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elasticsearch Service"
       }
   ],
   "ReturnSize": 6
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous GetDimensionValuesà la section Référence des AWS CLI commandes.

`get-reservation-coverage`

L'exemple de code suivant montre comment utiliser`get-reservation-coverage`.

AWS CLI

Pour récupérer la couverture des réservations pour les instances EC2 t2.nano dans la région us-east-1

Cet exemple permet de récupérer la couverture des réservations pour les instances EC2 t2.nano dans la région us-east-1 entre juillet et septembre 2017.

Commande :

```
aws ce get-reservation-coverage --time-period Start=2017-07-01,End=2017-10-01 --
group-by Type=Dimension,Key=REGION --filter file://filters.json
```

filtres.json :

```
{
   "And": [
     {
       "Dimensions": {
         "Key": "INSTANCE_TYPE",
         "Values": [
           "t2.nano"
         ]
       },
       "Dimensions": {
         "Key": "REGION",
         "Values": [
           "us-east-1"
         ]
       }
     }
   ]
```

```
 }
```

Sortie :

```
{
   "TotalSize": 6,
   "DimensionValues": [
       {
           "Attributes": {},
           "Value": "Amazon ElastiCache"
       },
       {
           "Attributes": {},
           "Value": "EC2 - Other"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Compute Cloud - Compute"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Load Balancing"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic MapReduce"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elasticsearch Service"
       }
   ],
   "ReturnSize": 6
}
```

- Pour plus de détails sur l'API, reportez-vous GetReservationCoverageà la section Référence des AWS CLI commandes.

`get-reservation-purchase-recommendation`

L'exemple de code suivant montre comment utiliser`get-reservation-purchase-recommendation`.

AWS CLI

Pour récupérer les recommandations de réservation pour une réservation initiale partielle d' EC2 RIs une durée de trois ans

L'`get-reservation-purchase-recommendation`exemple suivant extrait des recommandations pour les EC2 instances partielles initiales d'une durée de trois ans, sur la base des 60 derniers jours d' EC2 utilisation.

```
aws ce get-reservation-purchase-recommendation \
    --service "Amazon Redshift" \
    --lookback-period-in-days SIXTY_DAYS \
    --term-in-years THREE_YEARS \
    --payment-option PARTIAL_UPFRONT
```

Sortie :

```
{
    "Recommendations": [],
    "Metadata": {
        "GenerationTimestamp": "2018-08-08T15:20:57Z",
        "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetReservationPurchaseRecommendationà la section Référence des AWS CLI commandes.

`get-reservation-utilization`

L'exemple de code suivant montre comment utiliser`get-reservation-utilization`.

AWS CLI

Pour récupérer l'utilisation des réservations pour votre compte

L'`get-reservation-utilization`exemple suivant récupère l'utilisation du RI pour tous les types d'instances t2.nano entre le 01/03/2018 et le 01/08/2018 pour le compte.

```
aws ce get-reservation-utilization \
    --time-period Start=2018-03-01,End=2018-08-01 \
```

```
    --filter file://filters.json
```

Contenu de `filters.json` :

```
{
    "Dimensions": {
        "Key": "INSTANCE_TYPE",
        "Values": [
            "t2.nano"
        ]
    }
}
```

Sortie :

```
{
    "Total": {
        "TotalAmortizedFee": "0",
        "UtilizationPercentage": "0",
        "PurchasedHours": "0",
        "NetRISavings": "0",
        "TotalActualHours": "0",
        "AmortizedRecurringFee": "0",
        "UnusedHours": "0",
        "TotalPotentialRISavings": "0",
        "OnDemandCostOfRIHoursUsed": "0",
        "AmortizedUpfrontFee": "0"
    },
    "UtilizationsByTime": []
}
```

- Pour plus de détails sur l'API, reportez-vous GetReservationUtilizationà la section Référence des AWS CLI commandes.

`get-tags`

L'exemple de code suivant montre comment utiliser`get-tags`.

AWS CLI

Pour récupérer les clés et les valeurs d'une balise de répartition des coûts

Cet exemple extrait toutes les balises de répartition des coûts dont la clé est « Project » et une valeur contenant « SecretProject ».

Commande :

```
aws ce get-tags --search-string secretProject --time-
period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

Sortie :

```
{
  "ReturnSize": 2,
  "Tags": [
    "secretProject1",
    "secretProject2"
  ],
  "TotalSize": 2
}
```

- Pour plus de détails sur l'API, reportez-vous GetTagsà la section Référence des AWS CLI commandes.

Exemples de Firehose utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Firehose.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`list-delivery-streams`

L'exemple de code suivant montre comment utiliser`list-delivery-streams`.

AWS CLI

Pour répertorier les flux de diffusion disponibles

L'`list-delivery-streams`exemple suivant répertorie les flux de diffusion disponibles dans votre AWS compte.

```
aws firehose list-delivery-streams
```

Sortie :

```
{
    "DeliveryStreamNames": [
        "my-stream"
    ],
    "HasMoreDeliveryStreams": false
}
```

Pour plus d'informations, consultez Création d'un flux de diffusion Amazon Kinesis Data Firehose dans le Guide du développeur Amazon Kinesis Data Firehose.

- Pour plus de détails sur l'API, reportez-vous ListDeliveryStreamsà la section Référence des AWS CLI commandes.

`put-record-batch`

L'exemple de code suivant montre comment utiliser`put-record-batch`.

AWS CLI

Pour écrire plusieurs enregistrements dans un flux

L'`put-record-batch`exemple suivant écrit trois enregistrements dans un flux. Les données sont codées au format Base64.

```
aws firehose put-record-batch \
    --delivery-stream-name my-stream \
    --records file://records.json
```

Contenu de `myfile.json` :

```
[
    {"Data": "Rmlyc3QgdGhpbmc="},
    {"Data": "U2Vjb25kIHRoaW5n"},
    {"Data": "VGhpcmQgdGhpbmc="}
]
```

Sortie :

```
{
    "FailedPutCount": 0,
    "Encrypted": false,
    "RequestResponses": [
        {
            "RecordId": "9D2OJ6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhYOaWKPSmZy/
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7GiixpOyLJkYZcae1i7HzlCEoy9LJhMr8EjDSi4Om/9Vc2uhwwuAtGE0XKpxJ2WD7ZRWtAnYlKAnv
        },
        {
            "RecordId": "jFirejqxCLlK5xjH/UNmlMVcjktEN76I7916X9PaZ
+PVaOSXDfU1WGOqEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GKmR5z90l6XKJ
+/+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP9OiHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
        },
        {
            "RecordId":
 "oy0amQ40o5Y2YV4vxzufdcMOOw6n3EPr3tpPJGoYVNKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3iie
DTBt3qBlmTj7Xq8SKVbO1S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/sOkQLe+qh0sYKXWl"
        }
    ]
}
```

Pour plus d'informations, consultez la section Envoi de données vers un flux de diffusion Amazon Kinesis Data Firehose dans le manuel du développeur Amazon Kinesis Data Firehose.

- Pour plus de détails sur l'API, reportez-vous PutRecordBatchà la section Référence des AWS CLI commandes.

put-record

L'exemple de code suivant montre comment utiliserput-record.

AWS CLI

Pour écrire un enregistrement dans un flux

L'put-recordexemple suivant écrit des données dans un flux. Les données sont codées au format Base64.

```
aws firehose put-record \
    --delivery-stream-name my-stream \
    --record '{"Data":"SGVsbG8gd29ybGQ="}'
```

Sortie :

```
{
    "RecordId": "RjB5K/nnoGFHqwTsZlNd/
TTqvjE8V5dsyXZTQn2JXrdpMTOwssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/
j1Xujv31iDhnPdaLw4BOyM9Amv7PcCuB2079RuM0NhoakbyUymlwY8yt20G8X2420wu1jlFafhci4erAt7QhDEvpwuK8
    "Encrypted": false
}
```

Pour plus d'informations, consultez la section Envoi de données vers un flux de diffusion Amazon Kinesis Data Firehose dans le manuel du développeur Amazon Kinesis Data Firehose.

- Pour plus de détails sur l'API, reportez-vous PutRecordà la section Référence des AWS CLI commandes.

Exemples d'Amazon Data Lifecycle Manager utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Data Lifecycle Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-default-role`

L'exemple de code suivant montre comment utiliser`create-default-role`.

AWS CLI

Pour créer le rôle IAM requis pour Amazon DLM

L'`dlm create-default-role`exemple suivant crée le rôle AWS DataLifecycleManagerDefaultRole par défaut pour la gestion des instantanés.

```
aws dlm create-default-role \
    --resource-type snapshot
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Rôles de service par défaut pour Amazon Data Lifecycle Manager dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous CreateDefaultRoleà la section Référence des AWS CLI commandes.

`create-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`create-lifecycle-policy`.

AWS CLI

Pour créer une politique de cycle de vie

L'`create-lifecycle-policy`exemple suivant crée une politique de cycle de vie qui crée un instantané quotidien des volumes à l'heure spécifiée. Les balises spécifiées sont ajoutées aux instantanés, et les balises sont également copiées depuis le volume et ajoutées aux instantanés. Si la création d'un nouvel instantané dépasse le nombre maximum spécifié, le cliché le plus ancien est supprimé.

```
aws dlm create-lifecycle-policy \
    --description "My first policy" \
    --state ENABLED \
    --execution-role-arn arn:aws:iam::12345678910:role/
AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

Contenu de `policyDetails.json` :

```
{
    "ResourceTypes": [
        "VOLUME"
    ],
    "TargetTags": [
        {
            "Key": "costCenter",
            "Value": "115"
        }
    ],
    "Schedules":[
        {
            "Name": "DailySnapshots",
            "CopyTags": true,
            "TagsToAdd": [
                {
                    "Key": "type",
                     "Value": "myDailySnapshot"
                }
            ],
            "CreateRule": {
                "Interval": 24,
                "IntervalUnit": "HOURS",
                "Times": [
                    "03:00"
                ]
            },
            "RetainRule": {
                "Count":5
            }
        }
    ]
}
```

Sortie :

```
{
    "PolicyId": "policy-0123456789abcdef0"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLifecyclePolicyà la section Référence des AWS CLI commandes.

`delete-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`delete-lifecycle-policy`.

AWS CLI

Pour supprimer une politique de cycle de vie

L'exemple suivant supprime la politique de cycle de vie spécifiée. :

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- Pour plus de détails sur l'API, reportez-vous DeleteLifecyclePolicyà la section Référence des AWS CLI commandes.

`get-lifecycle-policies`

L'exemple de code suivant montre comment utiliser`get-lifecycle-policies`.

AWS CLI

Pour obtenir un résumé de vos politiques relatives au cycle de vie

L'`get-lifecycle-policies`exemple suivant répertorie toutes vos politiques de cycle de vie.

```
aws dlm get-lifecycle-policies
```

Sortie :

```
{
```

```
    "Policies": [
        {
            "PolicyId": "policy-0123456789abcdef0",
            "Description": "My first policy",
            "State": "ENABLED"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLifecyclePoliciesà la section Référence des AWS CLI commandes.

`get-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`get-lifecycle-policy`.

AWS CLI

Pour décrire une politique de cycle de vie

L'`get-lifecycle-policy`exemple suivant affiche les détails de la politique de cycle de vie spécifiée.

```
aws dlm get-lifecycle-policy \
    --policy-id policy-0123456789abcdef0
```

Sortie :

```
{
    "Policy": {
        "PolicyId": "policy-0123456789abcdef0",
        "Description": "My policy",
        "State": "ENABLED",
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/
AWSDataLifecycleManagerDefaultRole",
        "DateCreated": "2019-08-08T17:45:42Z",
        "DateModified": "2019-08-08T17:45:42Z",
        "PolicyDetails": {
            "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
            "ResourceTypes": [
                "VOLUME"
```

```
            ],
            "TargetTags": [
              {
                  "Key": "costCenter",
                  "Value": "115"
              }
            ],
            "Schedules": [
              {
                  "Name": "DailySnapshots",
                  "CopyTags": true,
                  "TagsToAdd": [
                    {
                        "Key": "type",
                        "Value": "myDailySnapshot"
                    }
                  ],
                  "CreateRule": {
                    "Interval": 24,
                    "IntervalUnit": "HOURS",
                    "Times": [
                        "03:00"
                    ]
                  },
                  "RetainRule": {
                    "Count": 5
                  }
              }
            ]
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetLifecyclePolicyà la section Référence des AWS CLI commandes.

`update-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`update-lifecycle-policy`.

AWS CLI

Exemple 1 : pour activer une politique de cycle de vie

L'`update-lifecycle-policy`exemple suivant active la politique de cycle de vie spécifiée.

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0 \
    --state ENABLED
```

Exemple 2 : pour désactiver une politique de cycle de vie

L'`update-lifecycle-policy`exemple suivant désactive la politique de cycle de vie spécifiée.

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0 \
    --state DISABLED
```

Exemple 3 : pour mettre à jour les détails de la politique de cycle de vie

L'`update-lifecycle-policy`exemple suivant met à jour les balises cibles pour la politique de cycle de vie spécifiée.

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0
    --policy-details file://policyDetails.json
```

Contenu de `policyDetails.json`. Les autres informations non référencées dans ce fichier ne sont pas modifiées par la commande.

```
{
    "TargetTags": [
        {
            "Key": "costCenter",
            "Value": "120"
        },
        {
            "Key": "project",
            "Value": "lima"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateLifecyclePolicyà la section Référence des AWS CLI commandes.

AWS Data Pipeline exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Data Pipeline.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`activate-pipeline`

L'exemple de code suivant montre comment utiliser`activate-pipeline`.

AWS CLI

Pour activer un pipeline

Cet exemple active le pipeline spécifié :

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

Pour activer le pipeline à une date et à une heure spécifiques, utilisez la commande suivante :

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --start-
timestamp 2015-04-07T00:00:00Z
```

- Pour plus de détails sur l'API, reportez-vous ActivatePipelineà la section Référence des AWS CLI commandes.

`add-tags`

L'exemple de code suivant montre comment utiliser`add-tags`.

AWS CLI

Pour ajouter une balise à un pipeline

Cet exemple ajoute la balise spécifiée au pipeline spécifié :

```
aws datapipeline add-tags --pipeline-id df-00627471SOVYZEXAMPLE --
tags key=environment,value=production key=owner,value=sales
```

Pour afficher les balises, utilisez la commande describe-pipelines. Par exemple, les balises ajoutées dans l'exemple de commande apparaissent comme suit dans la sortie de describe-pipelines :

```
{
    ...
        "tags": [
            {
                "value": "production",
                "key": "environment"
            },
            {
                "value": "sales",
                "key": "owner"
            }
        ]
    ...
}
```

- Pour plus de détails sur l'API, reportez-vous AddTagsà la section Référence des AWS CLI commandes.

`create-pipeline`

L'exemple de code suivant montre comment utiliser`create-pipeline`.

AWS CLI

Pour créer un pipeline

Cet exemple crée un pipeline :

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

Voici un exemple de sortie :

```
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePipelineà la section Référence des AWS CLI commandes.

`deactivate-pipeline`

L'exemple de code suivant montre comment utiliser`deactivate-pipeline`.

AWS CLI

Pour désactiver un pipeline

Cet exemple désactive le pipeline spécifié :

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

Pour désactiver le pipeline uniquement une fois toutes les activités en cours d'exécution terminées, utilisez la commande suivante :

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --no-
cancel-active
```

- Pour plus de détails sur l'API, reportez-vous DeactivatePipelineà la section Référence des AWS CLI commandes.

`delete-pipeline`

L'exemple de code suivant montre comment utiliser`delete-pipeline`.

AWS CLI

Pour supprimer un pipeline

Cet exemple supprime le pipeline spécifié :

```
aws datapipeline delete-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous DeletePipelineà la section Référence des AWS CLI commandes.

`describe-pipelines`

L'exemple de code suivant montre comment utiliser`describe-pipelines`.

AWS CLI

Pour décrire vos pipelines

Cet exemple décrit le pipeline spécifié :

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471SOVYZEXAMPLE
```

Voici un exemple de sortie :

```
{
  "pipelineDescriptionList": [
      {
          "fields": [
              {
                  "stringValue": "PENDING",
                  "key": "@pipelineState"
              },
              {
                  "stringValue": "my-pipeline",
                  "key": "name"
              },
              {
                  "stringValue": "2015-04-07T16:05:58",
                  "key": "@creationTime"
              },
              {
                  "stringValue": "df-00627471SOVYZEXAMPLE",
                  "key": "@id"
              },
```

```
            {
                "stringValue": "123456789012",
                "key": "pipelineCreator"
            },
            {
                "stringValue": "PIPELINE",
                "key": "@sphere"
            },
            {
                "stringValue": "123456789012",
                "key": "@userId"
            },
            {
                "stringValue": "123456789012",
                "key": "@accountId"
            },
            {
                "stringValue": "my-pipeline-token",
                "key": "uniqueId"
            }
        ],
        "pipelineId": "df-00627471SOVYZEXAMPLE",
        "name": "my-pipeline",
        "tags": []
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePipelinesà la section Référence des AWS CLI commandes.

`get-pipeline-definition`

L'exemple de code suivant montre comment utiliser`get-pipeline-definition`.

AWS CLI

Pour obtenir une définition de pipeline

Cet exemple obtient la définition du pipeline pour le pipeline spécifié :

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE
```

Voici un exemple de sortie :

```
{
  "parameters": [
      {
          "type": "AWS::S3::ObjectKey",
          "id": "myS3OutputLoc",
          "description": "S3 output folder"
      },
      {
          "default": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/data",
          "type": "AWS::S3::ObjectKey",
          "id": "myS3InputLoc",
          "description": "S3 input folder"
      },
      {
          "default": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
 ${OUTPUT1_STAGING_DIR}/output.txt",
          "type": "String",
          "id": "myShellCmd",
          "description": "Shell command to run"
      }
  ],
  "objects": [
      {
          "type": "Ec2Resource",
          "terminateAfter": "20 Minutes",
          "instanceType": "t1.micro",
          "id": "EC2ResourceObj",
          "name": "EC2ResourceObj"
      },
      {
          "name": "Default",
          "failureAndRerunMode": "CASCADE",
          "resourceRole": "DataPipelineDefaultResourceRole",
          "schedule": {
              "ref": "DefaultSchedule"
          },
          "role": "DataPipelineDefaultRole",
          "scheduleType": "cron",
          "id": "Default"
      },
      {
```

```
          "directoryPath": "#{myS3OutputLoc}/#{format(@scheduledStartTime, 'YYYY-MM-
dd-HH-mm-ss')}",
          "type": "S3DataNode",
          "id": "S3OutputLocation",
          "name": "S3OutputLocation"
      },
      {
          "directoryPath": "#{myS3InputLoc}",
          "type": "S3DataNode",
          "id": "S3InputLocation",
          "name": "S3InputLocation"
      },
      {
          "startAt": "FIRST_ACTIVATION_DATE_TIME",
          "name": "Every 15 minutes",
          "period": "15 minutes",
          "occurrences": "4",
          "type": "Schedule",
          "id": "DefaultSchedule"
      },
      {
          "name": "ShellCommandActivityObj",
          "command": "#{myShellCmd}",
          "output": {
              "ref": "S3OutputLocation"
          },
          "input": {
              "ref": "S3InputLocation"
          },
          "stage": "true",
          "type": "ShellCommandActivity",
          "id": "ShellCommandActivityObj",
          "runsOn": {
              "ref": "EC2ResourceObj"
          }
      }
  ],
  "values": {
      "myS3OutputLoc": "s3://amzn-s3-demo-bucket/",
      "myS3InputLoc": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/
data",
      "myShellCmd": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
 ${OUTPUT1_STAGING_DIR}/output.txt"
  }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous GetPipelineDefinitionà la section Référence des AWS CLI commandes.

`list-pipelines`

L'exemple de code suivant montre comment utiliser`list-pipelines`.

AWS CLI

Pour répertorier vos pipelines

Cet exemple répertorie vos pipelines :

```
aws datapipeline list-pipelines
```

Voici un exemple de sortie :

```
{
  "pipelineIdList": [
      {
          "id": "df-00627471SOVYZEXAMPLE",
          "name": "my-pipeline"
      },
      {
          "id": "df-09028963KNVMREXAMPLE",
          "name": "ImportDDB"
      },
      {
          "id": "df-0870198233ZYVEXAMPLE",
          "name": "CrossRegionDDB"
      },
      {
          "id": "df-00189603TB4MZEXAMPLE",
          "name": "CopyRedshift"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPipelinesà la section Référence des AWS CLI commandes.

`list-runs`

L'exemple de code suivant montre comment utiliser`list-runs`.

AWS CLI

Exemple 1 : Pour répertorier les cycles de votre pipeline

L'`list-runs`exemple suivant répertorie les essais pour le pipeline spécifié.

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE
```

Sortie :

```
    Name                        Scheduled Start        Status                      ID
                                              Started                Ended

 ---------------------------------------------------------------------------------------
1.  EC2ResourceObj              2015-04-12T17:33:02    CREATING
 @EC2ResourceObj_2015-04-12T17:33:02             2015-04-12T17:33:10
2.  S3InputLocation             2015-04-12T17:33:02    FINISHED
      @S3InputLocation_2015-04-12T17:33:02            2015-04-12T17:33:09
 2015-04-12T17:33:09
3.  S3OutputLocation            2015-04-12T17:33:02    WAITING_ON_DEPENDENCIES
 @S3OutputLocation_2015-04-12T17:33:02           2015-04-12T17:33:09
4.  ShellCommandActivityObj     2015-04-12T17:33:02    WAITING_FOR_RUNNER
 @ShellCommandActivityObj_2015-04-12T17:33:02    2015-04-12T17:33:09
```

Exemple 2 : Pour répertorier les cycles de pipeline entre les dates spécifiées

L'`list-runs`exemple suivant utilise le `--start-interval` pour spécifier les dates à inclure dans la sortie.

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUKO5 --start-
interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- Pour plus de détails sur l'API, reportez-vous ListRunsà la section Référence des AWS CLI commandes.

`put-pipeline-definition`

L'exemple de code suivant montre comment utiliser`put-pipeline-definition`.

AWS CLI

Pour télécharger une définition de pipeline

Cet exemple télécharge la définition de pipeline spécifiée vers le pipeline spécifié :

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://my-pipeline-definition.json
```

Voici un exemple de sortie :

```
{
  "validationErrors": [],
  "errored": false,
  "validationWarnings": []
}
```

- Pour plus de détails sur l'API, reportez-vous PutPipelineDefinitionà la section Référence des AWS CLI commandes.

`remove-tags`

L'exemple de code suivant montre comment utiliser`remove-tags`.

AWS CLI

Pour supprimer une balise d'un pipeline

Cet exemple supprime la balise spécifiée du pipeline spécifié :

```
aws datapipeline remove-tags --pipeline-id df-00627471SOVYZEXAMPLE --tag-
keys environment
```

- Pour plus de détails sur l'API, reportez-vous RemoveTagsà la section Référence des AWS CLI commandes.

DataSync exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with DataSync.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`update-location-azure-blob`

L'exemple de code suivant montre comment utiliser`update-location-azure-blob`.

AWS CLI

Pour mettre à jour le lieu de votre transfert avec un nouvel agent

L'`update-location-object-storage`exemple suivant met à jour votre DataSync position pour Microsoft Azure Blob Storage avec un nouvel agent.

```
aws datasync update-location-azure-blob \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --sas-configuration '{ \
        "Token": "sas-token-for-azure-blob-storage-access" \
    }'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Remplacement de votre agent dans le guide de AWS DataSync l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateLocationAzureBlobà la section Référence des AWS CLI commandes.

`update-location-hdfs`

L'exemple de code suivant montre comment utiliser`update-location-hdfs`.

AWS CLI

Pour mettre à jour le lieu de votre transfert avec un nouvel agent

L'`update-location-hdfs`exemple suivant met à jour votre emplacement DataSync HDFS avec un nouvel agent. Vous n'avez besoin des `--kerberos-krb5-conf` options `--kerberos-keytab` et que si votre cluster HDFS utilise l'authentification Kerberos.

```
aws datasync update-location-hdfs \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/agent-1234567890abcdef0 \
    --kerberos-keytab file://hdfs.keytab
    --kerberos-krb5-conf file://krb5.conf
```

Contenu de `hdfs.keytab` :

```
N/A. The content of this file is encrypted and not human readable.
```

Contenu de `krb5.conf` :

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = true
    ticket_lifetime = 24h
    forwardable = true
    udp_preference_limit = 1000000
    default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1
    default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1
    permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1
```

```
[realms]
    EXAMPLE.COM = {
      kdc = kdc1.example.com
      admin_server = krbadmin.example.com
      default_domain = example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM

[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kerberos/kadmin.log
    default = FILE:/var/log/krb5libs.log
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Remplacement de votre agent dans le guide de AWS DataSync l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateLocationHdfsà la section Référence des AWS CLI commandes.

`update-location-nfs`

L'exemple de code suivant montre comment utiliser`update-location-nfs`.

AWS CLI

Pour mettre à jour le lieu de votre transfert avec un nouvel agent

L'`update-location-nfs`exemple suivant met à jour votre emplacement DataSync NFS avec un nouvel agent.

```
aws datasync update-location-nfs \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --on-prem-config AgentArns=arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Remplacement de votre agent dans le guide de AWS DataSync l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateLocationNfsà la section Référence des AWS CLI commandes.

`update-location-object-storage`

L'exemple de code suivant montre comment utiliser`update-location-object-storage`.

AWS CLI

Pour mettre à jour le lieu de votre transfert avec un nouvel agent

L'`update-location-object-storage`exemple suivant met à jour votre emplacement de stockage d' DataSync objets avec un nouvel agent.

```
aws datasync update-location-object-storage \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --secret-key secret-key-for-object-storage
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Remplacement de votre agent dans le guide de AWS DataSync l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateLocationObjectStorageà la section Référence des AWS CLI commandes.

`update-location-smb`

L'exemple de code suivant montre comment utiliser`update-location-smb`.

AWS CLI

Pour mettre à jour le lieu de votre transfert avec un nouvel agent

L'`update-location-smb`exemple suivant met à jour l'emplacement de votre DataSync PME avec un nouvel agent.

```
aws datasync update-location-smb \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --password smb-file-server-password
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Remplacement de votre agent dans le guide de AWS DataSync l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateLocationSmbà la section Référence des AWS CLI commandes.

Exemples DAX utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants en utilisant le AWS Command Line Interface avec DAX.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-cluster`

L'exemple de code suivant montre comment utiliser`create-cluster`.

AWS CLI

Pour créer un cluster DAX

L'`create-cluster`exemple suivant crée un cluster DAX avec les paramètres spécifiés.

```
aws dax create-cluster \
    --cluster-name daxcluster \
    --node-type dax.r4.large \
    --replication-factor 3 \
    --iam-role-arn roleARN  \
    --sse-specification Enabled=true
```

Sortie :

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 0,
        "NodeType": "dax.r4.large",
        "Status": "creating",
        "ClusterDiscoveryEndpoint": {
            "Port": 8111
        },
        "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

Pour plus d'informations, consultez Étape 3 : Création d'un cluster DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous CreateClusterà la section Référence des AWS CLI commandes.

`create-parameter-group`

L'exemple de code suivant montre comment utiliser`create-parameter-group`.

AWS CLI

Pour créer un groupe de paramètres

L'exemple `` create-parameter-group`` suivant crée un groupe de paramètres avec les paramètres spécifiés.

```
aws dax create-parameter-group \
    --parameter-group-name daxparametergroup \
    --description "A new parameter group"
```

Sortie :

```
{
    "ParameterGroup": {
        "ParameterGroupName": "daxparametergroup",
        "Description": "A new parameter group"
    }
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous CreateParameterGroupà la section Référence des AWS CLI commandes.

`create-subnet-group`

L'exemple de code suivant montre comment utiliser`create-subnet-group`.

AWS CLI

Pour créer un groupe de sous-réseaux DAX

L'`create-subnet-group`exemple suivant crée un groupe de sous-réseaux avec les paramètres spécifiés.

```
aws dax create-subnet-group \
    --subnet-group-name daxSubnetGroup \
    --subnet-ids subnet-11111111 subnet-22222222
```

Sortie :

```
{
    "SubnetGroup": {
        "SubnetGroupName": "daxSubnetGroup",
        "VpcId": "vpc-05a1fa8e00c325226",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-11111111",
                "SubnetAvailabilityZone": "us-west-2b"
            },
            {
                "SubnetIdentifier": "subnet-22222222",
                "SubnetAvailabilityZone": "us-west-2c"
            }
        ]
    }
}
```

Pour plus d'informations, consultez Étape 2 : Création d'un groupe de sous-réseaux dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous CreateSubnetGroupà la section Référence des AWS CLI commandes.

`decrease-replication-factor`

L'exemple de code suivant montre comment utiliser`decrease-replication-factor`.

AWS CLI

Pour supprimer un ou plusieurs nœuds du cluster

L'`decrease-replication-factor`exemple suivant réduit à un le nombre de nœuds du cluster DAX spécifié.

```
aws dax decrease-replication-factor \
    --cluster-name daxcluster \
    --new-replication-factor 1
```

Sortie :

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 3,
        "NodeType": "dax.r4.large",
        "Status": "modifying",
        "ClusterDiscoveryEndpoint": {
            "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
            "Port": 8111
        },
        "Nodes": [
            {
                "NodeId": "daxcluster-a",
                "Endpoint": {
                    "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2c",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            },
            {
                "NodeId": "daxcluster-b",
                "Endpoint": {
                    "Address": "daxcluster-
b.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
```

```
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2a",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            },
            {
                "NodeId": "daxcluster-c",
                "Endpoint": {
                    "Address": "daxcluster-
c.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2b",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            }
        ],
        "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DecreaseReplicationFactorà la section Référence des AWS CLI commandes.

`delete-cluster`

L'exemple de code suivant montre comment utiliser`delete-cluster`.

AWS CLI

Pour supprimer un cluster DAX

L'`delete-cluster`exemple suivant supprime le cluster DAX spécifié.

```
aws dax delete-cluster \
    --cluster-name daxcluster
```

Sortie :

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 0,
        "NodeType": "dax.r4.large",
        "Status": "deleting",
        "ClusterDiscoveryEndpoint": {
            "Address": "dd.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
            "Port": 8111
        },
        "PreferredMaintenanceWindow": "fri:06:00-fri:07:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
```

```
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DeleteClusterà la section Référence des AWS CLI commandes.

`delete-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-parameter-group`.

AWS CLI

Pour supprimer un groupe de paramètres

L'`delete-parameter-group`exemple suivant supprime le groupe de paramètres DAX spécifié.

```
aws dax  delete-parameter-group \
    --parameter-group-name daxparametergroup
```

Sortie :

```
{
    "DeletionMessage": "Parameter group daxparametergroup has been deleted."
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DeleteParameterGroupà la section Référence des AWS CLI commandes.

`delete-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-subnet-group`.

AWS CLI

Pour supprimer un groupe de sous-réseaux

L'`delete-subnet-group`exemple suivant supprime le groupe de sous-réseaux DAX spécifié.

```
aws dax delete-subnet-group \
    --subnet-group-name daxSubnetGroup
```

Sortie :

```
{
    "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DeleteSubnetGroupà la section Référence des AWS CLI commandes.

`describe-clusters`

L'exemple de code suivant montre comment utiliser`describe-clusters`.

AWS CLI

Pour renvoyer des informations sur tous les clusters DAX provisionnés

L'`describe-clusters`exemple suivant affiche les détails de tous les clusters DAX provisionnés.

```
aws dax describe-clusters
```

Sortie :

```
{
    "Clusters": [
        {
            "ClusterName": "daxcluster",
            "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
            "TotalNodes": 1,
            "ActiveNodes": 1,
```

```
            "NodeType": "dax.r4.large",
            "Status": "available",
            "ClusterDiscoveryEndpoint": {
                "Address":
 "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
                "Port": 8111
            },
            "Nodes": [
                {
                    "NodeId": "daxcluster-a",
                    "Endpoint": {
                        "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                        "Port": 8111
                    },
                    "NodeCreateTime": 1576625059.509,
                    "AvailabilityZone": "us-west-2c",
                    "NodeStatus": "available",
                    "ParameterGroupStatus": "in-sync"
                }
            ],
            "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
            "SubnetGroup": "default",
            "SecurityGroups": [
                {
                    "SecurityGroupIdentifier": "sg-1af6e36e",
                    "Status": "active"
                }
            ],
            "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
            "ParameterGroup": {
                "ParameterGroupName": "default.dax1.0",
                "ParameterApplyStatus": "in-sync",
                "NodeIdsToReboot": []
            },
            "SSEDescription": {
                "Status": "ENABLED"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeClustersà la section Référence des AWS CLI commandes.

`describe-default-parameters`

L'exemple de code suivant montre comment utiliser`describe-default-parameters`.

AWS CLI

Pour renvoyer les informations de paramètres système par défaut pour DAX

L'`describe-default-parameters`exemple suivant affiche les informations des paramètres système par défaut pour DAX.

```
aws dax describe-default-parameters
```

Sortie :

```
{
    "Parameters": [
        {
            "ParameterName": "query-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for queries to remain cached",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        },
        {
            "ParameterName": "record-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for records to remain valid in
 cache (Default: 0 = infinite)",
            "Source": "user",
```

```
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeDefaultParametersà la section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour renvoyer tous les événements liés aux clusters DAX et aux groupes de paramètres

L'`describe-events`exemple suivant affiche les détails des événements liés aux clusters DAX et aux groupes de paramètres.

```
aws dax describe-events
```

Sortie :

```
{
    "Events": [
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Cluster deleted.",
            "Date": 1576702736.706
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Removed node daxcluster-b.",
            "Date": 1576702691.738
```

```
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Removed node daxcluster-a.",
            "Date": 1576702633.498
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Removed node daxcluster-c.",
            "Date": 1576702631.329
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Cluster created.",
            "Date": 1576626560.057
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeEventsà la section Référence des AWS CLI commandes.

`describe-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-parameter-groups`.

AWS CLI

Pour décrire les groupes de paramètres définis dans DAX

L'`describe-parameter-groups`exemple suivant permet de récupérer des informations sur les groupes de paramètres définis dans DAX.

```
aws dax describe-parameter-groups
```

Sortie :

```
{
    "ParameterGroups": [
        {
            "ParameterGroupName": "default.dax1.0",
            "Description": "Default parameter group for dax1.0"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeParameterGroupsà la section Référence des AWS CLI commandes.

`describe-parameters`

L'exemple de code suivant montre comment utiliser`describe-parameters`.

AWS CLI

Pour décrire les paramètres définis dans un groupe de paramètres DAX

L'`describe-parameters`exemple suivant permet de récupérer des informations sur les paramètres définis dans le groupe de paramètres DAX spécifié.

```
aws dax describe-parameters \
    --parameter-group-name default.dax1.0
```

Sortie :

```
{
    "Parameters": [
        {
            "ParameterName": "query-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for queries to remain cached",
            "Source": "user",
            "DataType": "integer",
```

```
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        },
        {
            "ParameterName": "record-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for records to remain valid in
 cache (Default: 0 = infinite)",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeParametersà la section Référence des AWS CLI commandes.

`describe-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-subnet-groups`.

AWS CLI

Pour décrire les groupes de sous-réseaux définis dans DAX

L'`describe-subnet-groups`exemple suivant récupère les détails des groupes de sous-réseaux définis dans DAX.

```
aws dax describe-subnet-groups
```

Sortie :

```
{
```

```
    "SubnetGroups": [
        {
            "SubnetGroupName": "default",
            "Description": "Default CacheSubnetGroup",
            "VpcId": "vpc-ee70a196",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-874953af",
                    "SubnetAvailabilityZone": "us-west-2d"
                },
                {
                    "SubnetIdentifier": "subnet-bd3d1fc4",
                    "SubnetAvailabilityZone": "us-west-2a"
                },
                {
                    "SubnetIdentifier": "subnet-72c2ff28",
                    "SubnetAvailabilityZone": "us-west-2c"
                },
                {
                    "SubnetIdentifier": "subnet-09e6aa42",
                    "SubnetAvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeSubnetGroupsà la section Référence des AWS CLI commandes.

`increase-replication-factor`

L'exemple de code suivant montre comment utiliser`increase-replication-factor`.

AWS CLI

Pour augmenter le facteur de réplication d'un cluster DAX

L'`increase-replication-factor`exemple suivant augmente le facteur de réplication du cluster DAX spécifié à 3.

```
aws dax increase-replication-factor \
    --cluster-name daxcluster \
    --new-replication-factor 3
```

Sortie :

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 1,
        "NodeType": "dax.r4.large",
        "Status": "modifying",
        "ClusterDiscoveryEndpoint": {
            "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
            "Port": 8111
        },
        "Nodes": [
            {
                "NodeId": "daxcluster-a",
                "Endpoint": {
                    "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2c",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            },
            {
                "NodeId": "daxcluster-b",
                "NodeStatus": "creating"
            },
            {
                "NodeId": "daxcluster-c",
                "NodeStatus": "creating"
            }
        ],
        "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
```

```
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous IncreaseReplicationFactorà la section Référence des AWS CLI commandes.

`list-tags`

L'exemple de code suivant montre comment utiliser`list-tags`.

AWS CLI

Pour répertorier les balises d'une ressource DAX

L'`list-tags`exemple suivant répertorie les clés de balise et les valeurs associées au cluster DAX spécifié.

```
aws dax list-tags \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

Sortie :

```
{
    "Tags": [
```

```
        {
            "Key": "ClusterUsage",
            "Value": "prod"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous ListTagsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour baliser une ressource DAX

L'`tag-resource`exemple suivant attache le nom de clé de balise spécifié et la valeur associée au cluster DAX spécifié pour décrire l'utilisation du cluster.

```
aws dax tag-resource \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
    --tags="Key=ClusterUsage,Value=prod"
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "ClusterUsage",
            "Value": "prod"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource DAX

L'`untag-resource`exemple suivant supprime la balise portant le nom de clé spécifié d'un cluster DAX.

```
aws dax untag-resource  \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
    --tag-keys="ClusterUsage"
```

Sortie :

```
{
    "Tags": []
}
```

Pour plus d'informations, consultez la section Gestion des clusters DAX dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples de Detective utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Detective.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-invitation`

L'exemple de code suivant montre comment utiliser`accept-invitation`.

AWS CLI

Pour accepter une invitation à devenir un compte membre dans un graphe de comportement

L'`accept-invitation`exemple suivant accepte une invitation à devenir un compte membre dans le graphe comportemental arn:aws:detective:us-east-1:111122223333:graph:123412341234.

```
aws detective accept-invitation \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Répondre à une invitation à un graphe de comportement dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous AcceptInvitationà la section Référence des AWS CLI commandes.

`create-graph`

L'exemple de code suivant montre comment utiliser`create-graph`.

AWS CLI

Pour activer Amazon Detective et créer un nouveau graphe de comportement

L'`create-graph`exemple suivant active Detective pour le AWS compte qui exécute la commande dans la région où la commande est exécutée. Un nouveau graphe de comportement est créé avec ce compte comme compte administrateur. La commande affecte également la valeur Finance à la balise Department.

```
aws detective create-graph \
    --tags '{"Department": "Finance"}'
```

Sortie :

```
{
    "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"
}
```

Pour plus d'informations, consultez la section Activation d'Amazon Detective dans le guide d'administration Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous CreateGraphà la section Référence des AWS CLI commandes.

`create-members`

L'exemple de code suivant montre comment utiliser`create-members`.

AWS CLI

Pour inviter les comptes des membres à accéder à un graphique de comportement

L'`create-members`exemple suivant invite deux AWS comptes à devenir des comptes membres dans le graphe de comportement arn:aws:detective:us-east-1:111122223333:graph:123412341234. Pour chaque compte, la demande fournit l'ID du AWS compte et l'adresse e-mail de l'utilisateur root du compte. La demande inclut un message personnalisé à insérer dans l'e-mail d'invitation.

```
aws detective create-members \
    --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --message "This is Paul Santos. I need to add your account to the data we use
 for security investigation in Amazon Detective. If you have any questions, contact
 me at psantos@example.com."
```

Sortie :

```
{
    "Members": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
   },
   {
        "AccountId": "123456789012",
        "AdministratorId": "111122223333",
        "EmailAddress": "jstiles@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "VERIFICATION_IN_PROGRESS",
        "UpdatedTime": 1579826107000
     }
    ],
    "UnprocessedAccounts": [ ]
}
```


Pour plus d'informations, consultez la section Invitation des comptes membres à un graphe de comportement < https://docs.aws.amazon.com/detective/ latest/adminguide/graph - admin-add-member-accounts .html> dans le guide d'administration d'Amazon Detective.

Pour inviter des comptes membres sans envoyer d'e-mails d'invitation

L'`create-members`exemple suivant invite deux AWS comptes à devenir des comptes membres dans le graphe de comportement arn:aws:detective:us-east-1:111122223333:graph:123412341234. Pour chaque compte, la demande fournit l'ID du AWS compte et l'adresse e-mail de l'utilisateur root du compte. Les comptes des membres ne reçoivent pas d'e-mails d'invitation.

```
aws detective create-members \
    --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
 \
```

```
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --disable-email-notification
```

Sortie :

```
{
    "Members": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
   },
   {
        "AccountId": "123456789012",
        "AdministratorId": "111122223333",
        "EmailAddress": "jstiles@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "VERIFICATION_IN_PROGRESS",
        "UpdatedTime": 1579826107000
     }
    ],
    "UnprocessedAccounts": [ ]
}
```

Pour plus d'informations, consultez la section Invitation des comptes membres à un graphe de comportement < https://docs.aws.amazon.com/detective/ latest/adminguide/graph - admin-add-member-accounts .html> dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous CreateMembersà la section Référence des AWS CLI commandes.

`delete-graph`

L'exemple de code suivant montre comment utiliser`delete-graph`.

AWS CLI

Pour désactiver Detective et supprimer le graphe de comportement

L'`delete-graph`exemple suivant désactive Detective et supprime le graphe de comportement spécifié.

```
aws detective delete-graph \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation d'Amazon Detective dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous DeleteGraphà la section Référence des AWS CLI commandes.

`delete-members`

L'exemple de code suivant montre comment utiliser`delete-members`.

AWS CLI

Pour supprimer des comptes de membres d'un graphique de comportement

L'`delete-members`exemple suivant supprime deux comptes membres du graphe de comportement arn:aws:detective:us-east- 1:111122223333:graph:123412341234. Pour identifier les comptes, la demande fournit le AWS compte IDs.

```
aws detective delete-members \
    --account-ids 444455556666 123456789012 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Sortie :

```
 {
    "AccountIds": [ "444455556666", "123456789012" ],
    "UnprocessedAccounts": [ ]
}
```

Pour plus d'informations, consultez Supprimer des comptes membres d'un graphe de comportement < https://docs.aws.amazon.com/detective/ latest/adminguide/graph - admin-remove-member-accounts .html> dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous DeleteMembersà la section Référence des AWS CLI commandes.

`disassociate-membership`

L'exemple de code suivant montre comment utiliser`disassociate-membership`.

AWS CLI

Pour résilier l'adhésion à partir d'un graphe de comportement

L'exemple de désassociation suivant supprime le AWS compte qui exécute la commande du graphe de comportement arn:aws:detective:us-east- 1:111122223333:graph:123412341234.

```
aws detective disassociate-membership \
     --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Pour plus d'informations, consultez Supprimer votre compte d'un graphe de comportement < https://docs.aws.amazon.com/detective/ latest/adminguide/member - remove-self-from-graph .html> dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous DisassociateMembershipà la section Référence des AWS CLI commandes.

`get-members`

L'exemple de code suivant montre comment utiliser`get-members`.

AWS CLI

Pour récupérer des informations sur les comptes membres du graphe comportemental sélectionnés

L'`get-members`exemple suivant extrait des informations sur deux comptes membres dans le graphe de comportement arn:aws:detective:us-east- 1:111122223333:graph:123412341234. Pour les deux comptes, la demande fournit le AWS compte IDs.

```
aws detective get-members \
    --account-ids 444455556666 123456789012 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Sortie :

```
{
    "MemberDetails": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
    }
    {
        "AccountId": "123456789012",
        "AdministratorId": "111122223333",
        "EmailAddress": "jstiles@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
    }
],
    "UnprocessedAccounts": [ ]
}
```

Pour plus d'informations, consultez la section Affichage de la liste des comptes dans un graphe de comportement < https://docs.aws.amazon.com/detective/ latest/adminguide/graph - admin-view-accounts .html> dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous GetMembersà la section Référence des AWS CLI commandes.

`list-graphs`

L'exemple de code suivant montre comment utiliser`list-graphs`.

AWS CLI

Pour consulter la liste des graphiques de comportement dont votre compte est l'administrateur

L'`list-graphs`exemple suivant récupère les graphiques de comportement dont le compte d'appel est l'administrateur dans la région actuelle.

```
aws detective list-graphs
```

Sortie :

```
{
    "GraphList": [
        {
            "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
            "CreatedTime": 1579736111000
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGraphsà la section Référence des AWS CLI commandes.

`list-invitations`

L'exemple de code suivant montre comment utiliser`list-invitations`.

AWS CLI

Pour consulter la liste des graphiques de comportement auxquels un compte est membre ou auquel il est invité

L'`list-invitations`exemple suivant permet de récupérer les graphiques de comportement auxquels le compte appelant a été invité. Les résultats incluent uniquement les invitations ouvertes et acceptées. Ils n'incluent pas les invitations rejetées ou les adhésions supprimées.

```
aws detective list-invitations
```

Sortie :

```
{
    "Invitations": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
    }
]
}
```

Pour plus d'informations, consultez la section Affichage de votre liste d'invitations à des graphes de comportement < https://docs.aws.amazon.com/detective/ latest/adminguide/member - view-graph-invitations .html> dans le guide d'administration d'Amazon Detective.

- Pour plus de détails sur l'API, reportez-vous ListInvitationsà la section Référence des AWS CLI commandes.

`list-members`

L'exemple de code suivant montre comment utiliser`list-members`.

AWS CLI

Pour répertorier les comptes des membres dans un graphique de comportement

L'`list-members`exemple suivant extrait les comptes des membres invités et activés pour le graphe `arn:aws:detective:us-east-1:111122223333:graph:123412341234` de comportement. Les résultats n'incluent pas les comptes de membres qui ont été supprimés.

```
aws detective list-members \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Sortie :

```
{
```

```
    "MemberDetails": [
        {
            "AccountId": "444455556666",
            "AdministratorId": "111122223333",
            "EmailAddress": "mmajor@example.com",
            "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:123412341234",
            "InvitedTime": 1579826107000,
            "MasterId": "111122223333",
            "Status": "INVITED",
            "UpdatedTime": 1579826107000
        },
        {
            "AccountId": "123456789012",
            "AdministratorId": "111122223333",
            "EmailAddress": "jstiles@example.com",
            "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:123412341234",
            "InvitedTime": 1579826107000,
            "MasterId": "111122223333",
            "PercentOfGraphUtilization": 2,
            "PercentOfGraphUtilizationUpdatedTime": 1586287843,
            "Status": "ENABLED",
            "UpdatedTime": 1579973711000,
            "VolumeUsageInBytes": 200,
            "VolumeUsageUpdatedTime": 1586287843
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage de la liste des comptes dans un graphique de comportement du Amazon Detective Administration Guide.

- Pour plus de détails sur l'API, reportez-vous ListMembersà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour récupérer les balises attribuées à un graphe de comportement

L'`list-tags-for-resource`exemple suivant renvoie les balises attribuées au graphe de comportement spécifié.

```
aws detective list-tags-for-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Sortie :

```
{
    "Tags": {
        "Department" : "Finance"
    }
}
```

Pour plus d'informations, consultez la section Gestion des balises pour un graphe de comportement dans le manuel Amazon Detective Administration Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`reject-invitation`

L'exemple de code suivant montre comment utiliser`reject-invitation`.

AWS CLI

Pour rejeter une invitation à devenir un compte membre dans un graphe de comportement

L'`reject-invitation`exemple suivant rejette une invitation à devenir un compte membre dans le graphe de comportement arn:aws:detective:us-east- 1:111122223333:graph:123412341234.

```
aws detective reject-invitation \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Responding to a behavior graph invitation< https:// docs.aws.amazon.com/detective/ latest/adminguide/member -invitation-response.html> dans le manuel Amazon Detective Administration Guide.

- Pour plus de détails sur l'API, reportez-vous RejectInvitationà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour attribuer un tag à une ressource

L'`tag-resource`exemple suivant attribue une valeur pour la balise Department au graphe de comportement spécifié.

```
aws detective tag-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --tags '{"Department":"Finance"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises pour un graphe de comportement dans le manuel Amazon Detective Administration Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une valeur de balise d'une ressource

L'`untag-resource`exemple suivant supprime la balise Department du graphe de comportement spécifié.

```
aws detective untag-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --tag-keys "Department"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises pour un graphe de comportement dans le manuel Amazon Detective Administration Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Device Farm AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Device Farm.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-device-pool`

L'exemple de code suivant montre comment utiliser`create-device-pool`.

AWS CLI

Pour créer un pool d'appareils

La commande suivante crée un pool d'appareils Android pour un projet :

```
aws devicefarm create-device-pool --name pool1 --rules file://
device-pool-rules.json --project-arn "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506"
```

Vous pouvez obtenir l'ARN du projet à partir de la sortie de `create-project` ou`list-projects`. Le fichier `device-pool-rules.json` est un document JSON situé dans le dossier actuel qui indique la plate-forme de l'appareil :

```
[
```

```
    {
        "attribute": "PLATFORM",
        "operator": "EQUALS",
        "value": "\"ANDROID\""
    }
]
```

Sortie :

```
{
    "devicePool": {
        "rules": [
            {
                "operator": "EQUALS",
                "attribute": "PLATFORM",
                "value": "\"ANDROID\""
            }
        ],
        "type": "PRIVATE",
        "name": "pool1",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:devicepool:070fc3ca-7ec1-4741-9c1f-
d3e044efc506/2aa8d2a9-5e73-47ca-b929-659cb34b7dcd"
    }
}
```

- Pour plus de détails sur l'API, voir CreateDevicePoolla section Référence des AWS CLI commandes.

`create-project`

L'exemple de code suivant montre comment utiliser`create-project`.

AWS CLI

Pour créer un projet

La commande suivante crée un nouveau projet nommé `my-project` :

```
aws devicefarm create-project --name my-project
```

Sortie :

```
{
    "project": {
        "name": "myproject",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
        "created": 1503612890.057
    }
}
```

- Pour plus de détails sur l'API, voir CreateProjectla section Référence des AWS CLI commandes.

`create-upload`

L'exemple de code suivant montre comment utiliser`create-upload`.

AWS CLI

Pour créer un téléchargement

La commande suivante crée un téléchargement pour une application Android :

```
aws devicefarm create-upload --project-arn "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506" --name app.apk --
type ANDROID_APP
```

Vous pouvez obtenir l'ARN du projet à partir de la sortie de create-project ou list-projects.

Sortie :

```
{
    "upload": {
        "status": "INITIALIZED",
        "name": "app.apk",
        "created": 1503614408.769,
        "url": "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
```

```
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f",
        "type": "ANDROID_APP",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
    }
}
```

Utilisez l'URL signée dans la sortie pour télécharger un fichier sur Device Farm :

```
curl -T app.apk "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f"
```

- Pour plus de détails sur l'API, voir CreateUploadla section Référence des AWS CLI commandes.

`get-upload`

L'exemple de code suivant montre comment utiliser`get-upload`.

AWS CLI

Pour consulter un téléchargement

La commande suivante permet de récupérer les informations relatives à un téléchargement :

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
```

Vous pouvez obtenir l'ARN de téléchargement à partir de la sortie de`create-upload`.

Sortie :

```
{
```

```
    "upload": {
        "status": "SUCCEEDED",
        "name": "app.apk",
        "created": 1505262773.186,
        "type": "ANDROID_APP",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514",
        "metadata": "{\"device_admin\":false,\"activity_name\":
\"ccom.example.client.LauncherActivity\",\"version_name\":\"1.0.2.94\",\"screens
\":[\"small\",\"normal\",\"large\",\"xlarge\"],\"error_type\":null,\"sdk_version
\":\"16\",\"package_name\":\"com.example.client\",\"version_code\":\"20994\",
\"native_code\":[\"armeabi-v7a\"],\"target_sdk_version\":\"25\"}"
    }
}
```

- Pour plus de détails sur l'API, voir GetUploadla section Référence des AWS CLI commandes.

`list-projects`

L'exemple de code suivant montre comment utiliser`list-projects`.

AWS CLI

Pour répertorier les projets

Voici une liste de projets :

```
aws devicefarm list-projects
```

Sortie :

```
{
    "projects": [
        {
            "name": "myproject",
            "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
            "created": 1503612890.057
        },
        {
            "name": "otherproject",
```

```
            "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:a5f5b752-8098-49d1-86bf-5f7682c1c77e",
            "created": 1505257519.337
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListProjectsla section Référence des AWS CLI commandes.

AWS Direct Connect exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Direct Connect.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-direct-connect-gateway-association-proposal`

L'exemple de code suivant montre comment utiliser`accept-direct-connect-gateway-association-proposal`.

AWS CLI

Pour accepter une proposition d'association de passerelle

Ce qui suit `accept-direct-connect-gateway-association-proposal` accepte la proposition spécifiée.

```
aws directconnect  accept-direct-connect-gateway-association-proposal \
```

```
    --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
    --proposal-id cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE \
    --associated-gateway-owner-account 111122223333

{
    "directConnectGatewayAssociation": {
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "directConnectGatewayOwnerAccount": "111122223333",
        "associationState": "associating",
        "associatedGateway": {
            "id": "tgw-02f776b1a7EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "111122223333",
            "region": "us-east-1"
        },
        "associationId": "6441f8bf-5917-4279-ade1-9708bEXAMPLE",
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.168.1.0/30"
            }
        ]

    }
}
```

Pour plus d'informations, voir Accepter ou rejeter une proposition d'association de Transit Gateway dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous AcceptDirectConnectGatewayAssociationProposalà la section Référence des AWS CLI commandes.

`allocate-connection-on-interconnect`

L'exemple de code suivant montre comment utiliser`allocate-connection-on-interconnect`.

AWS CLI

Pour créer une connexion hébergée sur une interconnexion

La `allocate-connection-on-interconnect` commande suivante crée une connexion hébergée sur une interconnexion :

```
aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101
```

Sortie :

```
{
    "partnerName": "TIVIT",
    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
}
```

- Pour plus de détails sur l'API, reportez-vous AllocateConnectionOnInterconnectà la section Référence des AWS CLI commandes.

`allocate-hosted-connection`

L'exemple de code suivant montre comment utiliser`allocate-hosted-connection`.

AWS CLI

Pour créer une connexion hébergée sur une interconnexion

L'`allocate-hosted-connection`exemple suivant crée une connexion hébergée sur l'interconnexion spécifiée.

```
aws directconnect allocate-hosted-connection \
    --bandwidth 500Mbps \
    --connection-name mydcinterconnect \
    --owner-account 123456789012
    -connection-id dxcon-fgktov66
    -vlan 101
```

Sortie :

```
{
    "partnerName": "TIVIT",
    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
}
```

- Pour plus de détails sur l'API, reportez-vous AllocateHostedConnectionà la section Référence des AWS CLI commandes.

`allocate-private-virtual-interface`

L'exemple de code suivant montre comment utiliser`allocate-private-virtual-interface`.

AWS CLI

Pour configurer une interface virtuelle privée

La `allocate-private-virtual-interface` commande suivante fournit une interface virtuelle privée qui appartiendra à un autre client :

```
aws directconnect allocate-private-virtual-interface --connection-id dxcon-
ffjrkx17 --owner-account 123456789012 --new-private-virtual-interface-
allocation virtualInterfaceName=PrivateVirtualInterface,vlan=1000,asn=65000,authKey=asdf34ex
```

Sortie :

```
{
    "virtualInterfaceState": "confirming",
    "asn": 65000,
    "vlan": 1000,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualInterfaceId": "dxvif-fgy8orxu",
    "authKey": "asdf34example",
```

```
    "routeFilterPrefixes": [],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n  <logical_connection id=\"dxvif-fgy8orxu\">\n  <vlan>1000</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "192.168.1.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "PrivateVirtualInterface"
}
```

- Pour plus de détails sur l'API, reportez-vous AllocatePrivateVirtualInterfaceà la section Référence des AWS CLI commandes.

`allocate-public-virtual-interface`

L'exemple de code suivant montre comment utiliser`allocate-public-virtual-interface`.

AWS CLI

Pour configurer une interface virtuelle publique

La `allocate-public-virtual-interface` commande suivante fournit une interface virtuelle publique qui appartiendra à un autre client :

```
aws directconnect allocate-public-virtual-interface --connection-id dxcon-
ffjrkx17 --owner-account 123456789012 --new-public-virtual-interface-
allocation virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exa
{cidr=203.0.113.4/30}]
```

Sortie :

```
{
    "virtualInterfaceState": "confirming",
    "asn": 65000,
    "vlan": 2000,
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
```

```
    "virtualInterfaceId": "dxvif-fg9xo9vp",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
        {
            "cidr": "203.0.113.0/30"
        },
        {
            "cidr": "203.0.113.4/30"
        }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fg9xo9vp\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
}
```

- Pour plus de détails sur l'API, reportez-vous AllocatePublicVirtualInterfaceà la section Référence des AWS CLI commandes.

`allocate-transit-virtual-interface`

L'exemple de code suivant montre comment utiliser`allocate-transit-virtual-interface`.

AWS CLI

Pour configurer une interface virtuelle de transit devant appartenir au AWS compte spécifié

L'`allocate-transit-virtual-interface`exemple suivant fournit une interface virtuelle de transit pour le compte spécifié.

```
aws directconnect allocate-transit-virtual-interface \
    --connection-id dxlag-fEXAMPLE \
    --owner-account 123456789012 \
    --new-transit-virtual-interface-allocation "virtualInterfaceName=Example Transit
 Virtual
 Interface,vlan=126,asn=65110,mtu=1500,authKey=0xzxgA9YoW9h58u8SEXAMPLE,amazonAddress=192.16
```

Sortie :


```
{
    "virtualInterface": {
        "ownerAccount": "123456789012",
        "virtualInterfaceId": "dxvif-fEXAMPLE",
        "location": "loc1",
        "connectionId": "dxlag-fEXAMPLE",
        "virtualInterfaceType": "transit",
        "virtualInterfaceName": "Example Transit Virtual Interface",
        "vlan": 126,
        "asn": 65110,
        "amazonSideAsn": 7224,
        "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "addressFamily": "ipv4",
        "virtualInterfaceState": "confirming",
        "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>126</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65110</bgp_asn>\n
 <bgp_auth_key>0xzxgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
        "mtu": 1500,
        "jumboFrameCapable": true,
        "virtualGatewayId": "",
        "directConnectGatewayId": "",
        "routeFilterPrefixes": [],
        "bgpPeers": [
            {
                "bgpPeerId": "dxpeer-fEXAMPLE",
                "asn": 65110,
                "authKey": "0xzxgA9YoW9h58u8EXAMPLE",
                "addressFamily": "ipv4",
                "amazonAddress": "192.168.1.1/30",
                "customerAddress": "192.168.1.2/30",
                "bgpPeerState": "pending",
                "bgpStatus": "down",
                "awsDeviceV2": "loc1-26wz6vEXAMPLE"
            }
        ],
        "region": "sa-east-1",
```

```
        "awsDeviceV2": "loc1-26wz6vEXAMPLE",
        "tags": [
            {
                "key": "Tag",
                "value": "Example"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'une interface virtuelle de transit hébergée dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous AllocateTransitVirtualInterfaceà la section Référence des AWS CLI commandes.

`associate-connection-with-lag`

L'exemple de code suivant montre comment utiliser`associate-connection-with-lag`.

AWS CLI

Pour associer une connexion à un LAG

L'exemple suivant associe la connexion spécifiée au LAG spécifié.

Commande :

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --
connection-id dxcon-fg9607vm
```

Sortie :

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateConnectionWithLagà la section Référence des AWS CLI commandes.

`associate-hosted-connection`

L'exemple de code suivant montre comment utiliser`associate-hosted-connection`.

AWS CLI

Pour associer une connexion hébergée à un LAG

L'exemple suivant associe la connexion hébergée spécifiée au LAG spécifié.

Commande :

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t
 --connection-id dxcon-fg9607vm
```

Sortie :

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateHostedConnectionà la section Référence des AWS CLI commandes.

`associate-virtual-interface`

L'exemple de code suivant montre comment utiliser`associate-virtual-interface`.

AWS CLI

Pour associer une interface virtuelle à une connexion

L'exemple suivant associe l'interface virtuelle spécifiée au LAG spécifié. Sinon, pour associer l'interface virtuelle à une connexion, spécifiez l'ID d'une connexion AWS Direct Connect pour `--connection-id` ; par exemple,`dxcon-ffnikghc`.

Commande :

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --
virtual-interface-id dxvif-fgputw0j
```

Sortie :

```
{
  "virtualInterfaceState": "pending",
  "asn": 65000,
  "vlan": 123,
  "customerAddress": "169.254.255.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxlag-ffjhj9lx",
  "addressFamily": "ipv4",
  "virtualGatewayId": "vgw-38e90b51",
  "virtualInterfaceId": "dxvif-fgputw0j",
  "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
  "routeFilterPrefixes": [],
  "location": "CSVA1",
  "bgpPeers": [
      {
          "bgpStatus": "down",
          "customerAddress": "169.254.255.2/30",
          "addressFamily": "ipv4",
          "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
          "bgpPeerState": "deleting",
          "amazonAddress": "169.254.255.1/30",
          "asn": 65000
      },
      {
          "bgpStatus": "down",
          "customerAddress": "169.254.255.2/30",
          "addressFamily": "ipv4",
          "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
```

```
            "bgpPeerState": "pending",
            "amazonAddress": "169.254.255.1/30",
            "asn": 65000
        }
    ],
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgputw0j\">\n  <vlan>123</vlan>
\n  <customer_address>169.254.255.2/30</customer_address>\n
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0x123pK5_VBqv.UQ3kJ4123_</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "VIF1A"
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateVirtualInterfaceà la section Référence des AWS CLI commandes.

`confirm-connection`

L'exemple de code suivant montre comment utiliser`confirm-connection`.

AWS CLI

Pour confirmer la création d'une connexion hébergée sur une interconnexion

La `confirm-connection` commande suivante confirme la création d'une connexion hébergée sur une interconnexion :

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

Sortie :

```
{
    "connectionState": "pending"
}
```

- Pour plus de détails sur l'API, reportez-vous ConfirmConnectionà la section Référence des AWS CLI commandes.

`confirm-private-virtual-interface`

L'exemple de code suivant montre comment utiliser`confirm-private-virtual-interface`.

AWS CLI

Pour accepter la propriété d'une interface virtuelle privée

La `confirm-private-virtual-interface` commande suivante accepte la propriété d'une interface virtuelle privée créée par un autre client :

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

Sortie :

```
{
    "virtualInterfaceState": "pending"
}
```

- Pour plus de détails sur l'API, reportez-vous ConfirmPrivateVirtualInterfaceà la section Référence des AWS CLI commandes.

`confirm-public-virtual-interface`

L'exemple de code suivant montre comment utiliser`confirm-public-virtual-interface`.

AWS CLI

Pour accepter la propriété d'une interface virtuelle publique

La `confirm-public-virtual-interface` commande suivante accepte la propriété d'une interface virtuelle publique créée par un autre client :

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

Sortie :

```
{
```

```
    "virtualInterfaceState": "verifying"
}
```

- Pour plus de détails sur l'API, reportez-vous ConfirmPublicVirtualInterfaceà la section Référence des AWS CLI commandes.

`confirm-transit-virtual-interface`

L'exemple de code suivant montre comment utiliser`confirm-transit-virtual-interface`.

AWS CLI

Pour accepter la propriété d'une interface virtuelle de transport

Ce qui suit `confirm-transit-virtual-interface` accepte la propriété d'une interface virtuelle de transport créée par un autre client.

```
aws directconnect confirm-transit-virtual-interface \
    --virtual-interface-id dxvif-fEXAMPLE \
    --direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```

Sortie :

```
{
    "virtualInterfaceState": "pending"
}
```

Pour plus d'informations, consultez la section Acceptation d'une interface virtuelle hébergée dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous ConfirmTransitVirtualInterfaceà la section Référence des AWS CLI commandes.

`create-bgp-peer`

L'exemple de code suivant montre comment utiliser`create-bgp-peer`.

AWS CLI

Pour créer une session de IPv6 peering BGP

L'exemple suivant crée une session d'appairage IPv6 BGP sur une interface virtuelle privée. `dxvif-fg1vuj3d` Les IPv6 adresses homologues sont automatiquement attribuées par Amazon.

Commande :

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

Sortie :

```
{
  "virtualInterface": {
      "virtualInterfaceState": "available",
      "asn": 65000,
      "vlan": 125,
      "customerAddress": "169.254.255.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-fguhmqlc",
      "addressFamily": "ipv4",
      "virtualGatewayId": "vgw-f9eb0c90",
      "virtualInterfaceId": "dxvif-fg1vuj3d",
      "authKey": "0xC_ukbCerl6EYA0example",
      "routeFilterPrefixes": [],
      "location": "EqDC2",
      "bgpPeers": [
          {
              "bgpStatus": "down",
              "customerAddress": "169.254.255.2/30",
              "addressFamily": "ipv4",
              "authKey": "0xC_ukbCerl6EYA0uexample",
              "bgpPeerState": "available",
              "amazonAddress": "169.254.255.1/30",
              "asn": 65000
          },
          {
              "bgpStatus": "down",
              "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
              "addressFamily": "ipv6",
              "authKey": "0xS27kAIU_VHPjjAexample",
              "bgpPeerState": "pending",
              "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
              "asn": 64600
          }
```

```
      ],
      "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</
bgp_asn>\n  <bgp_auth_key>0xC_ukbCerl6EYA0uexample</bgp_auth_key>\n
 <ipv6_customer_address>2001:db8:1100:2f0:0:1:9cb4:4216/125</ipv6_customer_address>
\n  <ipv6_amazon_address>2001:db8:1100:2f0:0:1:9cb4:4211/125</ipv6_amazon_address>\n
  <ipv6_bgp_asn>64600</ipv6_bgp_asn>\n  <ipv6_bgp_auth_key>0xS27kAIU_VHPjjAexample</
ipv6_bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
 <connection_type>private</connection_type>\n</logical_connection>\n",
      "amazonAddress": "169.254.255.1/30",
      "virtualInterfaceType": "private",
      "virtualInterfaceName": "Test"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateBgpPeerà la section Référence des AWS CLI commandes.

`create-connection`

L'exemple de code suivant montre comment utiliser`create-connection`.

AWS CLI

Pour créer une connexion entre votre réseau et un AWS point Direct Connect

La `create-connection` commande suivante crée une connexion entre votre réseau et un emplacement AWS Direct Connect :

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-
name "Connection to AWS"
```

Sortie :

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
```

```
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateConnectionà la section Référence des AWS CLI commandes.

`create-direct-connect-gateway-association-proposal`

L'exemple de code suivant montre comment utiliser`create-direct-connect-gateway-association-proposal`.

AWS CLI

Pour créer une proposition visant à associer la passerelle de transit spécifiée à la passerelle Direct Connect spécifiée

L'`create-direct-connect-gateway-association-proposal`exemple suivant crée une proposition qui associe la passerelle de transit spécifiée à la passerelle Direct Connect spécifiée.

```
aws directconnect create-direct-connect-gateway-association-proposal \
    --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
    --direct-connect-gateway-owner-account 111122223333 \
    --gateway-id tgw-02f776b1a7EXAMPLE \
    --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.1.0/30
```

Sortie :

```
{
    "directConnectGatewayAssociationProposal": {
        "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "directConnectGatewayOwnerAccount": "111122223333",
        "proposalState": "requested",
        "associatedGateway": {
            "id": "tgw-02f776b1a7EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "111122223333",
            "region": "us-east-1"
        },
        "requestedAllowedPrefixesToDirectConnectGateway": [
```

```
            {
                "cidr": "192.168.1.0/30"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'une proposition d'association Transit Gateway dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous CreateDirectConnectGatewayAssociationProposalà la section Référence des AWS CLI commandes.

`create-direct-connect-gateway-association`

L'exemple de code suivant montre comment utiliser`create-direct-connect-gateway-association`.

AWS CLI

Pour associer une passerelle privée virtuelle à une passerelle Direct Connect

L'exemple suivant associe une passerelle privée virtuelle `vgw-6efe725e` à une passerelle Direct Connect`5f294f92-bafb-4011-916d-9b0bexample`. Vous devez exécuter la commande dans la région dans laquelle se trouve la passerelle privée virtuelle.

Commande :

```
aws directconnect create-direct-connect-gateway-association --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

Sortie :

```
{
  "directConnectGatewayAssociation": {
      "associationState": "associating",
      "virtualGatewayOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualGatewayId": "vgw-6efe725e",
      "virtualGatewayRegion": "us-east-2"
  }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDirectConnectGatewayAssociationà la section Référence des AWS CLI commandes.

`create-direct-connect-gateway`

L'exemple de code suivant montre comment utiliser`create-direct-connect-gateway`.

AWS CLI

Pour créer une passerelle Direct Connect

L'exemple suivant crée une passerelle Direct Connect portant le nom`DxGateway1`.

Commande :

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-
name "DxGateway1"
```

Sortie :

```
{
  "directConnectGateway": {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "available"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDirectConnectGatewayà la section Référence des AWS CLI commandes.

`create-interconnect`

L'exemple de code suivant montre comment utiliser`create-interconnect`.

AWS CLI

Pour créer une interconnexion entre le réseau d'un partenaire et AWS

La `create-interconnect` commande suivante crée une interconnexion entre le réseau d'un partenaire AWS Direct Connect et un emplacement AWS Direct Connect spécifique :

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
 --bandwidth 1Gbps --location TIVIT
```

Sortie :

```
{
    "region": "sa-east-1",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "interconnectName": "1G Interconnect to AWS",
    "interconnectId": "dxcon-fgktov66",
    "interconnectState": "requested"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateInterconnectà la section Référence des AWS CLI commandes.

`create-lag`

L'exemple de code suivant montre comment utiliser`create-lag`.

AWS CLI

Pour créer un LAG avec de nouvelles connexions

L'exemple suivant crée un LAG et demande deux nouvelles connexions AWS Direct Connect pour le LAG avec une bande passante de 1 Gbit/s.

Commande :

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 1GBLag
```

Sortie :

```
{
  "awsDevice": "CSVA1-23u8tlpaz8iks",
  "numberOfConnections": 2,
```

```
  "lagState": "pending",
  "ownerAccount": "123456789012",
  "lagName": "1GBLag",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-ffqr6x5q",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fflqyj95",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      }
  ],
  "lagId": "dxlag-ffjhj9lx",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "CSVA1"
}
```

Pour créer un LAG à l'aide d'une connexion existante

L'exemple suivant crée un LAG à partir d'une connexion existante dans votre compte et demande une deuxième nouvelle connexion pour le LAG avec la même bande passante et le même emplacement que la connexion existante.

Commande :

```
aws directconnect create-lag --location EqDC2 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dr
```

Sortie :

```
{
  "awsDevice": "EqDC2-4h6ce2r1bes6",
  "numberOfConnections": 2,
  "lagState": "pending",
  "ownerAccount": "123456789012",
  "lagName": "2ConnLAG",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fh6ljcvo",
          "lagId": "dxlag-fhccu14t",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "EqDC2",
          "connectionName": "Requested Connection 1 for Lag dxlag-fhccu14t",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fgk145dr",
          "lagId": "dxlag-fhccu14t",
          "connectionState": "down",
          "bandwidth": "1Gbps",
          "location": "EqDC2",
          "connectionName": "VAConn1",
          "region": "us-east-1"
      }
  ],
  "lagId": "dxlag-fhccu14t",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "EqDC2"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLagà la section Référence des AWS CLI commandes.

`create-private-virtual-interface`

L'exemple de code suivant montre comment utiliser`create-private-virtual-interface`.

AWS CLI

Pour créer une interface virtuelle privée

La `create-private-virtual-interface` commande suivante crée une interface virtuelle privée :

```
aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrkx17 --
new-private-virtual-
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam
aba37db6
```

Sortie :

```
{
    "virtualInterfaceState": "pending",
    "asn": 65000,
    "vlan": 101,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualGatewayId": "vgw-aba37db6",
    "virtualInterfaceId": "dxvif-ffhhk74f",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "192.168.1.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "PrivateVirtualInterface"
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePrivateVirtualInterfaceà la section Référence des AWS CLI commandes.

`create-public-virtual-interface`

L'exemple de code suivant montre comment utiliser`create-public-virtual-interface`.

AWS CLI

Pour créer une interface virtuelle publique

La `create-public-virtual-interface` commande suivante crée une interface virtuelle publique :

```
aws directconnect create-public-virtual-interface --connection-id dxcon-ffjrkx17 --
new-public-virtual-
interface virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exam
{cidr=203.0.113.4/30}]
```

Sortie :

```
{
    "virtualInterfaceState": "verifying",
    "asn": 65000,
    "vlan": 2000,
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualInterfaceId": "dxvif-fgh0hcrk",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
        {
            "cidr": "203.0.113.0/30"
        },
        {
            "cidr": "203.0.113.4/30"
        }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
```

```
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePublicVirtualInterfaceà la section Référence des AWS CLI commandes.

`create-transit-virtual-interface`

L'exemple de code suivant montre comment utiliser`create-transit-virtual-interface`.

AWS CLI

Pour créer une interface virtuelle de transport

L'`create-transit-virtual-interface`exemple suivant crée une interface virtuelle de transit pour la connexion spécifiée.

```
aws directconnect create-transit-virtual-interface \
    --connection-id dxlag-fEXAMPLE \
    --new-transit-virtual-interface "virtualInterfaceName=Example Transit Virtual
 Interface,vlan=126,asn=65110,mtu=1500,authKey=0xzxgA9YoW9h58u8SvEXAMPLE,amazonAddress=192.1
aada-5a1baEXAMPLE,tags=[{key=Tag,value=Example}]"
```

Sortie :

```
{
    "virtualInterface": {
        "ownerAccount": "1111222233333",
        "virtualInterfaceId": "dxvif-fEXAMPLE",
        "location": "loc1",
        "connectionId": "dxlag-fEXAMPLE",
        "virtualInterfaceType": "transit",
        "virtualInterfaceName": "Example Transit Virtual Interface",
        "vlan": 126,
        "asn": 65110,
        "amazonSideAsn": 4200000000,
        "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "addressFamily": "ipv4",
```

```
        "virtualInterfaceState": "pending",
        "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>126</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65110</
bgp_asn>\n  <bgp_auth_key>0xzxgA9YoW9h58u8SvOmXRTw</bgp_auth_key>\n
 <amazon_bgp_asn>4200000000</amazon_bgp_asn>\n  <connection_type>transit</
connection_type>\n</logical_connection>\n",
        "mtu": 1500,
        "jumboFrameCapable": true,
        "virtualGatewayId": "",
        "directConnectGatewayId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
        "routeFilterPrefixes": [],
        "bgpPeers": [
            {
                "bgpPeerId": "dxpeer-EXAMPLE",
                "asn": 65110,
                "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
                "addressFamily": "ipv4",
                "amazonAddress": "192.168.1.1/30",
                "customerAddress": "192.168.1.2/30",
                "bgpPeerState": "pending",
                "bgpStatus": "down",
                "awsDeviceV2": "loc1-26wz6vEXAMPLE"
            }
        ],
        "region": "sa-east-1",
        "awsDeviceV2": "loc1-26wz6vEXAMPLE",
        "tags": [
            {
                "key": "Tag",
                "value": "Example"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'une interface virtuelle de transit vers la passerelle Direct Connect dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous CreateTransitVirtualInterfaceà la section Référence des AWS CLI commandes.

`delete-bgp-peer`

L'exemple de code suivant montre comment utiliser`delete-bgp-peer`.

AWS CLI

Pour supprimer un homologue BGP d'une interface virtuelle

L'exemple suivant supprime l'homologue IPv6 BGP de l'interface virtuelle. `dxvif-fg1vuj3d`

Commande :

```
aws directconnect delete-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --asn 64600
 --customer-address 2001:db8:1100:2f0:0:1:9cb4:4216/125
```

Sortie :

```
{
  "virtualInterface": {
      "virtualInterfaceState": "available",
      "asn": 65000,
      "vlan": 125,
      "customerAddress": "169.254.255.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-fguhmqlc",
      "addressFamily": "ipv4",
      "virtualGatewayId": "vgw-f9eb0c90",
      "virtualInterfaceId": "dxvif-fg1vuj3d",
      "authKey": "0xC_ukbCerl6EYA0example",
      "routeFilterPrefixes": [],
      "location": "EqDC2",
      "bgpPeers": [
          {
              "bgpStatus": "down",
              "customerAddress": "169.254.255.2/30",
              "addressFamily": "ipv4",
              "authKey": "0xC_ukbCerl6EYA0uexample",
              "bgpPeerState": "available",
              "amazonAddress": "169.254.255.1/30",
              "asn": 65000
          },
          {
              "bgpStatus": "down",
```

```
            "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
            "addressFamily": "ipv6",
            "authKey": "0xS27kAIU_VHPjjAexample",
            "bgpPeerState": "deleting",
            "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
            "asn": 64600
        }
    ],
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0xC_ukbCerl6EYA0example</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "Test"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBgpPeerà la section Référence des AWS CLI commandes.

`delete-connection`

L'exemple de code suivant montre comment utiliser`delete-connection`.

AWS CLI

Pour supprimer une connexion

La `delete-connection` commande suivante supprime la connexion spécifiée :

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

Sortie :

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "connectionState": "deleted",
```

```
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteConnectionà la section Référence des AWS CLI commandes.

`delete-direct-connect-gateway-association`

L'exemple de code suivant montre comment utiliser`delete-direct-connect-gateway-association`.

AWS CLI

Pour supprimer une association de passerelle Direct Connect

L'`delete-direct-connect-gateway-association`exemple suivant supprime l'association de passerelle Direct Connect avec une passerelle de transit dont l'ID d'association est spécifié.

```
aws directconnect delete-direct-connect-gateway-association --association-id
  be85116d-46eb-4b43-a27a-da0c2ad648de
```

Sortie :

```
{
    "directConnectGatewayAssociation": {
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPlE",
        "directConnectGatewayOwnerAccount": "123456789012",
        "associationState": "disassociating",
        "associatedGateway": {
            "id": "tgw-095b3b0b54EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "123456789012",
            "region": "us-east-1"
       },
        "associationId": " be85116d-46eb-4b43-a27a-da0c2ad648deEXAMPLE ",
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.0.1.0/28"
```

```
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Association et dissociation des passerelles de transport dans le guide de l'utilisateur de Direct AWS Connect.

- Pour plus de détails sur l'API, reportez-vous DeleteDirectConnectGatewayAssociationà la section Référence des AWS CLI commandes.

`delete-direct-connect-gateway`

L'exemple de code suivant montre comment utiliser`delete-direct-connect-gateway`.

AWS CLI

Pour supprimer une passerelle Direct Connect

L'exemple suivant supprime la passerelle `5f294f92-bafb-4011-916d-9b0bexample` Direct Connect.

Commande :

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-
id 5f294f92-bafb-4011-916d-9b0bexample
```

Sortie :

```
{
  "directConnectGateway": {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "deleting"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDirectConnectGatewayà la section Référence des AWS CLI commandes.

`delete-interconnect`

L'exemple de code suivant montre comment utiliser`delete-interconnect`.

AWS CLI

Pour supprimer une interconnexion

La `delete-interconnect` commande suivante supprime l'interconnexion spécifiée :

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

Sortie :

```
{
    "interconnectState": "deleted"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteInterconnectà la section Référence des AWS CLI commandes.

`delete-lag`

L'exemple de code suivant montre comment utiliser`delete-lag`.

AWS CLI

Pour supprimer un LAG

L'exemple suivant supprime le LAG spécifié.

Commande :

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

Sortie :

```
{
  "awsDevice": "EqDC2-4h6ce2r1bes6",
  "numberOfConnections": 0,
  "lagState": "deleted",
```

```
  "ownerAccount": "123456789012",
  "lagName": "TestLAG",
  "connections": [],
  "lagId": "dxlag-ffrhowd9",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "EqDC2"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteLagà la section Référence des AWS CLI commandes.

`delete-virtual-interface`

L'exemple de code suivant montre comment utiliser`delete-virtual-interface`.

AWS CLI

Pour supprimer une interface virtuelle

La `delete-virtual-interface` commande suivante supprime l'interface virtuelle spécifiée :

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhhk74f
```

Sortie :

```
{
    "virtualInterfaceState": "deleting"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteVirtualInterfaceà la section Référence des AWS CLI commandes.

`describe-connection-loa`

L'exemple de code suivant montre comment utiliser`describe-connection-loa`.

AWS CLI

Pour décrire votre LOA-CFA pour une connexion utilisant Linux ou Mac OS X

L'exemple suivant décrit votre LOA-CFA pour la connexion. `dxcon-fh6ayh1d` Le contenu du LOA-CFA est codé en base64. Cette commande utilise les `--query` paramètres `--output` et pour contrôler la sortie et extraire le contenu de la `loaContent` structure. La dernière partie de la commande décode le contenu à l'aide de l'`base64`utilitaire et envoie le résultat dans un fichier PDF.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Pour décrire votre LOA-CFA pour une connexion utilisant Windows

L'exemple précédent nécessite l'utilisation de l'`base64`utilitaire pour décoder la sortie. Sur un ordinateur Windows, vous pouvez utiliser à la `certutil` place. Dans l'exemple suivant, la première commande décrit votre LOA-CFA pour la connexion `dxcon-fh6ayh1d` et utilise les `--query` paramètres `--output` et pour contrôler la sortie et extraire le contenu de la `loaContent` structure dans un fichier appelé. `myLoaCfa.base64` La deuxième commande utilise l'utilitaire `certutil` pour décoder le fichier et envoyer le résultat vers un fichier PDF.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Pour plus d'informations sur le contrôle de la sortie de la AWS CLI, consultez la section Contrôle de la sortie de AWS commande depuis l'interface de ligne de commande dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DescribeConnectionLoaà la section Référence des AWS CLI commandes.

`describe-connections-on-interconnect`

L'exemple de code suivant montre comment utiliser`describe-connections-on-interconnect`.

AWS CLI

Pour répertorier les connexions sur une interconnexion

La `describe-connections-on-interconnect` commande suivante répertorie les connexions qui ont été configurées sur l'interconnexion donnée :

```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-fgktov66
```

Sortie :

```
{
    "connections": [
        {
            "partnerName": "TIVIT",
            "vlan": 101,
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffzc51m1",
            "connectionState": "ordering",
            "bandwidth": "500Mbps",
            "location": "TIVIT",
            "connectionName": "mydcinterconnect",
            "region": "sa-east-1"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConnectionsOnInterconnectà la section Référence des AWS CLI commandes.

`describe-connections`

L'exemple de code suivant montre comment utiliser`describe-connections`.

AWS CLI

Pour répertorier toutes les connexions dans la région actuelle

La `describe-connections` commande suivante répertorie toutes les connexions dans la région actuelle :

```
aws directconnect describe-connections
```

Sortie :

```
{
  "connections": [
```

```
        {
            "awsDevice": "EqDC2-123h49s71dabc",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-fguhmqlc",
            "lagId": "dxlag-ffrz71kw",
            "connectionState": "down",
            "bandwidth": "1Gbps",
            "location": "EqDC2",
            "connectionName": "My_Connection",
            "loaIssueTime": 1491568964.0,
            "region": "us-east-1"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConnectionsà la section Référence des AWS CLI commandes.

`describe-direct-connect-gateway-association-proposals`

L'exemple de code suivant montre comment utiliser`describe-direct-connect-gateway-association-proposals`.

AWS CLI

Pour décrire vos propositions d'association de passerelles Direct Connect

L'`describe-direct-connect-gateway-association-proposals`exemple suivant affiche les détails de vos propositions d'association de passerelles Direct Connect.

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

Sortie :

```
{
    "directConnectGatewayAssociationProposals": [
        {
            "proposalId": "c2ede9b4-bbc6-4d33-923c-bc4feEXAMPLE",
            "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
            "directConnectGatewayOwnerAccount": "111122223333",
            "proposalState": "requested",
            "associatedGateway": {
```

```
                "id": "tgw-02f776b1a7EXAMPLE",
                "type": "transitGateway",
                "ownerAccount": "111122223333",
                "region": "us-east-1"
            },
            "existingAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.2.0/30"
                },
                {
                    "cidr": "192.168.1.0/30"
                }
            ],
            "requestedAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.1.0/30"
                }
            ]
        },
        {
            "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
            "directConnectGatewayId": "11560968-4ac1-4fd3-bcb2-00599EXAMPLE",
            "directConnectGatewayOwnerAccount": "111122223333",
            "proposalState": "accepted",
            "associatedGateway": {
                "id": "tgw-045776b1a7EXAMPLE",
                "type": "transitGateway",
                "ownerAccount": "111122223333",
                "region": "us-east-1"
            },
            "existingAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.4.0/30"
                },
                {
                    "cidr": "192.168.5.0/30"
                }
            ],
            "requestedAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.5.0/30"
                }
            ]
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Association et dissociation des passerelles de transport dans le guide de l'utilisateur de Direct AWS Connect.

- Pour plus de détails sur l'API, reportez-vous DescribeDirectConnectGatewayAssociationProposalsà la section Référence des AWS CLI commandes.

`describe-direct-connect-gateway-associations`

L'exemple de code suivant montre comment utiliser`describe-direct-connect-gateway-associations`.

AWS CLI

Pour décrire les associations de passerelles Direct Connect

L'exemple suivant décrit toutes les associations avec la passerelle Direct Connect`5f294f92-bafb-4011-916d-9b0bexample`.

Commande :

```
aws directconnect describe-direct-connect-gateway-associations --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Sortie :

```
{
  "nextToken":
 "eyJ2IjoxLCJzIjoxLCJpIjoiOU83OTFodzdycnZCbkN4MExHeHVwQT09IiwiYyI6InIxTEN0UEVHV0I1UFlkaWFnNl
  "directConnectGatewayAssociations": [
      {
          "associationState": "associating",
          "virtualGatewayOwnerAccount": "123456789012",
          "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
          "virtualGatewayId": "vgw-6efe725e",
          "virtualGatewayRegion": "us-east-2"
      },
      {
          "associationState": "disassociating",
```

```
            "virtualGatewayOwnerAccount": "123456789012",
            "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
            "virtualGatewayId": "vgw-ebaa27db",
            "virtualGatewayRegion": "us-east-2"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDirectConnectGatewayAssociationsà la section Référence des AWS CLI commandes.

`describe-direct-connect-gateway-attachments`

L'exemple de code suivant montre comment utiliser`describe-direct-connect-gateway-attachments`.

AWS CLI

Pour décrire les pièces jointes de la passerelle Direct Connect

L'exemple suivant décrit les interfaces virtuelles connectées à la passerelle Direct Connect`5f294f92-bafb-4011-916d-9b0bexample`.

Commande :

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Sortie :

```
{
    "directConnectGatewayAttachments": [
        {
            "virtualInterfaceOwnerAccount": "123456789012",
            "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
            "virtualInterfaceRegion": "us-east-2",
            "attachmentState": "attaching",
            "virtualInterfaceId": "dxvif-fg9zyabc"
        }
    ],
    "nextToken":
 "eyJ2IjoxLCJzIjoxLCJpIjoibEhXdlNpUXF5RzhoL1JyUW52SlV2QT09IiwiYyI6Im5wQjFHQ0RyQUdRS3puNnNXcU
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDirectConnectGatewayAttachmentsà la section Référence des AWS CLI commandes.

`describe-direct-connect-gateways`

L'exemple de code suivant montre comment utiliser`describe-direct-connect-gateways`.

AWS CLI

Pour décrire vos passerelles Direct Connect

L'exemple suivant décrit toutes vos passerelles Direct Connect.

Commande :

```
aws directconnect describe-direct-connect-gateways
```

Sortie :

```
{
  "directConnectGateways": [
      {
          "amazonSideAsn": 64512,
          "directConnectGatewayId": "cf68415c-f4ae-48f2-87a7-3b52cexample",
          "ownerAccount": "123456789012",
          "directConnectGatewayName": "DxGateway2",
          "directConnectGatewayState": "available"
      },
      {
          "amazonSideAsn": 64512,
          "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
          "ownerAccount": "123456789012",
          "directConnectGatewayName": "DxGateway1",
          "directConnectGatewayState": "available"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDirectConnectGatewaysà la section Référence des AWS CLI commandes.

`describe-hosted-connections`

L'exemple de code suivant montre comment utiliser`describe-hosted-connections`.

AWS CLI

Pour répertorier les connexions sur une interconnexion

L'exemple suivant répertorie les connexions qui ont été configurées sur l'interconnexion donnée.

Commande :

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

Sortie :

```
{
    "connections": [
        {
            "partnerName": "TIVIT",
            "vlan": 101,
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffzc51m1",
            "connectionState": "ordering",
            "bandwidth": "500Mbps",
            "location": "TIVIT",
            "connectionName": "mydcinterconnect",
            "region": "sa-east-1"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeHostedConnectionsà la section Référence des AWS CLI commandes.

`describe-interconnect-loa`

L'exemple de code suivant montre comment utiliser`describe-interconnect-loa`.

AWS CLI

Pour décrire votre LOA-CFA pour une interconnexion utilisant Linux ou Mac OS X

L'exemple suivant décrit votre LOA-CFA pour l'interconnexion. `dxcon-fh6ayh1d` Le contenu du LOA-CFA est codé en base64. Cette commande utilise les `--query` paramètres `--output` et pour contrôler la sortie et extraire le contenu de la `loaContent` structure. La dernière partie de la commande décode le contenu à l'aide de l'`base64`utilitaire et envoie le résultat dans un fichier PDF.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Pour décrire votre LOA-CFA pour une interconnexion utilisant Windows

L'exemple précédent nécessite l'utilisation de l'`base64`utilitaire pour décoder la sortie. Sur un ordinateur Windows, vous pouvez utiliser à la `certutil` place. Dans l'exemple suivant, la première commande décrit votre LOA-CFA pour l'interconnexion `dxcon-fh6ayh1d` et utilise les `--query` paramètres `--output` et pour contrôler la sortie et extraire le contenu de la `loaContent` structure dans un fichier appelé. `myLoaCfa.base64` La deuxième commande utilise l'utilitaire `certutil` pour décoder le fichier et envoyer le résultat vers un fichier PDF.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Pour plus d'informations sur le contrôle de la sortie de la AWS CLI, consultez la section Contrôle de la sortie de AWS commande depuis l'interface de ligne de commande dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DescribeInterconnectLoaà la section Référence des AWS CLI commandes.

`describe-interconnects`

L'exemple de code suivant montre comment utiliser`describe-interconnects`.

AWS CLI

Pour répertorier les interconnexions

La `describe-interconnects` commande suivante répertorie les interconnexions détenues par votre AWS compte :

```
aws directconnect describe-interconnects
```

Sortie :

```
{
    "interconnects": [
        {
            "region": "sa-east-1",
            "bandwidth": "1Gbps",
            "location": "TIVIT",
            "interconnectName": "1G Interconnect to AWS",
            "interconnectId": "dxcon-fgktov66",
            "interconnectState": "down"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInterconnectsà la section Référence des AWS CLI commandes.

`describe-lags`

L'exemple de code suivant montre comment utiliser`describe-lags`.

AWS CLI

Pour décrire votre LAGs

La commande suivante décrit tous vos paramètres LAGs pour la région en cours.

Commande :

```
aws directconnect describe-lags
```

Sortie :

```
{
  "lags": [
      {
          "awsDevice": "EqDC2-19y7z3m17xpuz",
          "numberOfConnections": 2,
          "lagState": "down",
```

```
            "ownerAccount": "123456789012",
            "lagName": "DA-LAG",
            "connections": [
                {
                    "ownerAccount": "123456789012",
                    "connectionId": "dxcon-ffnikghc",
                    "lagId": "dxlag-fgsu9erb",
                    "connectionState": "requested",
                    "bandwidth": "10Gbps",
                    "location": "EqDC2",
                    "connectionName": "Requested Connection 1 for Lag dxlag-fgsu9erb",
                    "region": "us-east-1"
                },
                {
                    "ownerAccount": "123456789012",
                    "connectionId": "dxcon-fglgbdea",
                    "lagId": "dxlag-fgsu9erb",
                    "connectionState": "requested",
                    "bandwidth": "10Gbps",
                    "location": "EqDC2",
                    "connectionName": "Requested Connection 2 for Lag dxlag-fgsu9erb",
                    "region": "us-east-1"
                }
            ],
            "lagId": "dxlag-fgsu9erb",
            "minimumLinks": 0,
            "connectionsBandwidth": "10Gbps",
            "region": "us-east-1",
            "location": "EqDC2"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLagsà la section Référence des AWS CLI commandes.

`describe-loa`

L'exemple de code suivant montre comment utiliser`describe-loa`.

AWS CLI

Pour décrire votre LOA-CFA pour une connexion utilisant Linux ou Mac OS X

L'exemple suivant décrit votre LOA-CFA pour la connexion. `dxcon-fh6ayh1d` Le contenu du LOA-CFA est codé en base64. Cette commande utilise les `--query` paramètres `--output` et pour contrôler la sortie et extraire le contenu de la `loaContent` structure. La dernière partie de la commande décode le contenu à l'aide de l'`base64`utilitaire et envoie le résultat dans un fichier PDF.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Pour décrire votre LOA-CFA pour une connexion utilisant Windows

L'exemple précédent nécessite l'utilisation de l'`base64`utilitaire pour décoder la sortie. Sur un ordinateur Windows, vous pouvez utiliser à la `certutil` place. Dans l'exemple suivant, la première commande décrit votre LOA-CFA pour la connexion `dxcon-fh6ayh1d` et utilise les `--query` paramètres `--output` et pour contrôler la sortie et extraire le contenu de la `loaContent` structure dans un fichier appelé. `myLoaCfa.base64` La deuxième commande utilise l'utilitaire `certutil` pour décoder le fichier et envoyer le résultat vers un fichier PDF.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Pour plus d'informations sur le contrôle de la sortie de la AWS CLI, consultez la section Contrôle de la sortie de AWS commande depuis l'interface de ligne de commande dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DescribeLoaà la section Référence des AWS CLI commandes.

`describe-locations`

L'exemple de code suivant montre comment utiliser`describe-locations`.

AWS CLI

Pour répertorier les partenaires et les sites AWS Direct Connect

La `describe-locations` commande suivante répertorie les partenaires et les sites AWS Direct Connect dans la région actuelle :

```
aws directconnect describe-locations
```

Sortie :

```
{
    "locations": [
        {
            "locationName": "NAP do Brasil, Barueri, Sao Paulo",
            "locationCode": "TNDB"
        },
        {
            "locationName": "Tivit - Site Transamerica (Sao Paulo)",
            "locationCode": "TIVIT"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLocationsà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire les balises associées à vos ressources AWS Direct Connect

La commande suivante décrit les balises de la connexion`dxcon-abcabc12`.

Commande :

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12
```

Sortie :

```
{
  "resourceTags": [
      {
```

```
          "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-
abcabc12",
          "tags": [
              {
                  "value": "VAConnection",
                  "key": "Name"
              }
          ]
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`describe-virtual-gateways`

L'exemple de code suivant montre comment utiliser`describe-virtual-gateways`.

AWS CLI

Pour répertorier les passerelles privées virtuelles

La `describe-virtual-gateways` commande suivante répertorie les passerelles privées virtuelles détenues par votre AWS compte :

```
aws directconnect describe-virtual-gateways
```

Sortie :

```
{
    "virtualGateways": [
        {
            "virtualGatewayId": "vgw-aba37db6",
            "virtualGatewayState": "available"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVirtualGatewaysà la section Référence des AWS CLI commandes.

`describe-virtual-interfaces`

L'exemple de code suivant montre comment utiliser`describe-virtual-interfaces`.

AWS CLI

Pour répertorier toutes les interfaces virtuelles

La `describe-virtual-interfaces` commande suivante répertorie les informations relatives à toutes les interfaces virtuelles associées à votre AWS compte :

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrkx17
```

Sortie :

```
{
    "virtualInterfaces": [
        {
            "virtualInterfaceState": "down",
            "asn": 65000,
            "vlan": 101,
            "customerAddress": "192.168.1.2/30",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffjrkx17",
            "virtualGatewayId": "vgw-aba37db6",
            "virtualInterfaceId": "dxvif-ffhhk74f",
            "authKey": "asdf34example",
            "routeFilterPrefixes": [],
            "location": "TIVIT",
            "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
            "amazonAddress": "192.168.1.1/30",
            "virtualInterfaceType": "private",
            "virtualInterfaceName": "PrivateVirtualInterface"
        },
        {
            "virtualInterfaceState": "verifying",
            "asn": 65000,
            "vlan": 2000,
```

```
            "customerAddress": "203.0.113.2/30",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffjrkx17",
            "virtualGatewayId": "",
            "virtualInterfaceId": "dxvif-fgh0hcrk",
            "authKey": "asdf34example",
            "routeFilterPrefixes": [
                {
                    "cidr": "203.0.113.4/30"
                },
                {
                    "cidr": "203.0.113.0/30"
                }
            ],
            "location": "TIVIT",
            "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
            "amazonAddress": "203.0.113.1/30",
            "virtualInterfaceType": "public",
            "virtualInterfaceName": "PublicVirtualInterface"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVirtualInterfacesà la section Référence des AWS CLI commandes.

`disassociate-connection-from-lag`

L'exemple de code suivant montre comment utiliser`disassociate-connection-from-lag`.

AWS CLI

Pour dissocier une connexion d'un LAG

L'exemple suivant dissocie la connexion spécifiée du LAG spécifié.

Commande :

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --
connection-id  dxcon-fg9607vm
```

Sortie :

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateConnectionFromLagà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter un tag à une ressource AWS Direct Connect

La commande suivante ajoute une balise avec une clé `Name` et une valeur de `VAConnection` à la connexion`dxcon-abcabc12`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource AWS Direct Connect

La commande suivante supprime la balise contenant la clé `Name` de la connexion`dxcon-abcabc12`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-direct-connect-gateway-association`

L'exemple de code suivant montre comment utiliser`update-direct-connect-gateway-association`.

AWS CLI

Pour mettre à jour les attributs spécifiés de l'association de passerelle Direct Connect

L'`update-direct-connect-gateway-association`exemple suivant ajoute le bloc CIDR spécifié à une association de passerelle Direct Connect.

```
aws directconnect update-direct-connect-gateway-association \
    --association-id 820a6e4f-5374-4004-8317-3f64bEXAMPLE \
    --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.2.0/30
```

Sortie :

```
{
    "directConnectGatewayAssociation": {
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "directConnectGatewayOwnerAccount": "111122223333",
        "associationState": "updating",
        "associatedGateway": {
            "id": "tgw-02f776b1a7EXAMPLE",
            "type": "transitGateway",
```

```
            "ownerAccount": "111122223333",
            "region": "us-east-1"
        },
        "associationId": "820a6e4f-5374-4004-8317-3f64bEXAMPLE",
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.168.2.0/30"
            },
            {
                "cidr": "192.168.1.0/30"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Utilisation des passerelles Direct Connect dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateDirectConnectGatewayAssociationà la section Référence des AWS CLI commandes.

`update-lag`

L'exemple de code suivant montre comment utiliser`update-lag`.

AWS CLI

Pour mettre à jour un LAG

L'exemple suivant modifie le nom du LAG spécifié.

Commande :

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

Sortie :

```
{
  "awsDevice": "CSVA1-23u8tlpaz8iks",
  "numberOfConnections": 2,
  "lagState": "down",
  "ownerAccount": "123456789012",
```

```
  "lagName": "2ConnLag",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fflqyj95",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-ffqr6x5q",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      }
  ],
  "lagId": "dxlag-ffjhj9lx",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "CSVA1"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateLagà la section Référence des AWS CLI commandes.

`update-virtual-interface-attributes`

L'exemple de code suivant montre comment utiliser`update-virtual-interface-attributes`.

AWS CLI

Pour mettre à jour le MTU d'une interface virtuelle

L'`update-virtual-interface-attributes`exemple suivant met à jour le MTU de l'interface virtuelle spécifiée.

```
aws directconnect update-virtual-interface-attributes \
    --virtual-interface-id dxvif-fEXAMPLE \
    --mtu 1500
```

Sortie :

```
{
    "ownerAccount": "1111222233333",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "example transit virtual interface",
    "vlan": 125,
    "asn": 650001,
    "amazonSideAsn": 64512,
    "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
    "amazonAddress": "169.254.248.1/30",
    "customerAddress": "169.254.248.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "down",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>125</vlan>
\n  <customer_address>169.254.248.2/30</customer_address>\n
 <amazon_address>169.254.248.1/30</amazon_address>\n  <bgp_asn>650001</bgp_asn>\n
  <bgp_auth_key>0xzxgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>64512</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
    "mtu": 1500,
    "jumboFrameCapable": true,
    "virtualGatewayId": "",
    "directConnectGatewayId": "879b76a1-403d-4700-8b53-4a56ed85436e",
    "routeFilterPrefixes": [],
    "bgpPeers": [
        {
            "bgpPeerId": "dxpeer-fEXAMPLE",
            "asn": 650001,
            "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
            "addressFamily": "ipv4",
            "amazonAddress": "169.254.248.1/30",
            "customerAddress": "169.254.248.2/30",
            "bgpPeerState": "available",
            "bgpStatus": "down",
```

```
            "awsDeviceV2": "loc1-26wz6vEXAMPLE"
        }
    ],
    "region": "sa-east-1",
    "awsDeviceV2": "loc1-26wz6vEXAMPLE",
    "tags": []
}
```

Pour plus d'informations, consultez la section Configuration de la MTU réseau pour les interfaces virtuelles privées ou les interfaces virtuelles de transit dans le guide de l'utilisateur de AWS Direct Connect.

- Pour plus de détails sur l'API, reportez-vous UpdateVirtualInterfaceAttributesà la section Référence des AWS CLI commandes.

AWS Directory Service exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Directory Service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-directories`

L'exemple de code suivant montre comment utiliser`describe-directories`.

AWS CLI

Pour obtenir des informations sur vos annuaires

L'`describe-directories`exemple suivant affiche les détails du répertoire spécifié.

```
aws ds describe-directories \
   --directory-id d-a1b2c3d4e5
```

Sortie :


```
{
    "DirectoryDescriptions": [
        {
            "DirectoryId": "d-a1b2c3d4e5",
            "Name": "mydirectory.example.com",
            "ShortName": "mydirectory",
            "Size": "Small",
            "Edition": "Standard",
            "Alias": "d-a1b2c3d4e5",
            "AccessUrl": "d-a1b2c3d4e5.awsapps.com",
            "Stage": "Active",
            "ShareStatus": "Shared",
            "ShareMethod": "HANDSHAKE",
            "ShareNotes": "These are my share notes",
            "LaunchTime": "2019-07-08T15:33:46.327000-07:00",
            "StageLastUpdatedDateTime": "2019-07-08T15:59:12.307000-07:00",
            "Type": "SharedMicrosoftAD",
            "SsoEnabled": false,
            "DesiredNumberOfDomainControllers": 0,
            "OwnerDirectoryDescription": {
                "DirectoryId": "d-b2c3d4e5f6",
                "AccountId": "123456789111",
                "DnsIpAddrs": [
                    "203.113.0.248",
                    "203.113.0.253"
                ],
                "VpcSettings": {
                    "VpcId": "vpc-a1b2c3d4",
                    "SubnetIds": [
                        "subnet-a1b2c3d4",
                        "subnet-d4c3b2a1"
                    ],
                    "AvailabilityZones": [
                        "us-west-2a",
                        "us-west-2c"
                    ]
                }
            }
```

```
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDirectoriesà la section Référence des AWS CLI commandes.

`describe-trusts`

L'exemple de code suivant montre comment utiliser`describe-trusts`.

AWS CLI

Pour obtenir des informations sur vos relations de confiance

L'`describe-trusts`exemple suivant affiche des détails sur les relations d'approbation pour le répertoire spécifié.

```
aws ds describe-trusts \
   --directory-id d-a1b2c3d4e5
```

Sortie :

```
{
    "Trusts": [
        {
            "DirectoryId": "d-a1b2c3d4e5",
            "TrustId": "t-9a8b7c6d5e",
            "RemoteDomainName": "other.example.com",
            "TrustType": "Forest",
            "TrustDirection": "Two-Way",
            "TrustState": "Verified",
            "CreatedDateTime": "2017-06-20T18:08:45.614000-07:00",
            "LastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",
            "StateLastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",
            "SelectiveAuth": "Disabled"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTrustsà la section Référence des AWS CLI commandes.

AWS Directory Service Exemples de données utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with AWS Directory Service Data.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-group-member`

L'exemple de code suivant montre comment utiliser`add-group-member`.

AWS CLI

Pour ajouter un membre du groupe à un annuaire

L'`add-group-member`exemple suivant ajoute l'utilisateur spécifié au groupe spécifié dans le répertoire spécifié.

```
aws ds-data add-group-member \
    --directory-id d-1234567890 \
    --group-name 'sales' \
    --member-name 'john.doe'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout ou suppression de membres Microsoft AD AWS gérés à des groupes et de groupes à des groupes dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous AddGroupMemberà la section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour répertorier les widgets disponibles

L'`create-group`exemple suivant crée un groupe dans un répertoire spécifié.

```
aws ds-data create-group \
    --directory-id d-1234567890 \
    --sam-account-name "sales"
```

Sortie :

```
{
    "DirectoryId": "d-1234567890",
    "SAMAccountName": "sales",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

Pour plus d'informations, consultez la section Création d'un groupe Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un utilisateur

L'`create-user`exemple suivant crée un utilisateur dans le répertoire spécifié.

```
aws ds-data create-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

Sortie :

```
{
    "DirectoryId": "d-1234567890",
    "SAMAccountName": "john.doe",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

Pour plus d'informations, consultez la section Création d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe

L'`delete-group`exemple suivant supprime le groupe spécifié du répertoire spécifié.

```
aws ds-data delete-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un groupe Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

L'`delete-user`exemple suivant supprime l'utilisateur spécifié du répertoire spécifié.

```
aws ds-data delete-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`describe-group`

L'exemple de code suivant montre comment utiliser`describe-group`.

AWS CLI

Pour répertorier les détails d'un groupe

L'`describe-group`exemple suivant permet d'obtenir des informations pour le groupe spécifié dans le répertoire spécifié.

```
aws ds-data describe-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

Sortie :

```
{
    "DirectoryId": "d-1234567890",
    "DistinguishedName": "CN=sales,OU=Users,OU=CORP,DC=corp,DC=example,DC=com",
    "GroupScope": "Global",
    "GroupType": "Security",
    "Realm": "corp.example.com",
    "SAMAccountName": "sales",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour des informations relatives à un groupe Microsoft AD AWS géré dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous DescribeGroupà la section Référence des AWS CLI commandes.

`describe-user`

L'exemple de code suivant montre comment utiliser`describe-user`.

AWS CLI

Pour répertorier les informations d'un utilisateur

L'`describe-user`exemple suivant permet d'obtenir des informations pour l'utilisateur spécifié dans le répertoire spécifié.

```
aws ds-data describe-user command-name \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

Sortie :

```
{
    "DirectoryId": "d-1234567890",
    "DistinguishedName": "CN=john.doe,OU=Users,OU=CORP,DC=corp,DC=example,DC=com",
    "Enabled": false,
    "Realm": "corp.example.com",
    "SAMAccountName": "john.doe",
    "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567",
    "UserPrincipalName": "john.doe@CORP.EXAMPLE.COM"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous DescribeUserà la section Référence des AWS CLI commandes.

`disable-directory-data-access`

L'exemple de code suivant montre comment utiliser`disable-directory-data-access`.

AWS CLI

Pour désactiver l'API Directory Service Data pour un annuaire

L'`disable-directory-data-access`exemple suivant désactive l'API Directory Service Data pour le répertoire spécifié.

```
aws ds disable-directory-data-access \
    --directory-id d-1234567890
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation ou désactivation de la gestion des utilisateurs et des groupes ou des données des services d' AWS annuaire dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous DisableDirectoryDataAccessà la section Référence des AWS CLI commandes.

`disable-user`

L'exemple de code suivant montre comment utiliser`disable-user`.

AWS CLI

Pour désactiver un utilisateur

L'`disable-user`exemple suivant désactive l'utilisateur spécifié dans le répertoire spécifié.

```
aws ds-data disable-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous DisableUserà la section Référence des AWS CLI commandes.

`enable-directory-data-access`

L'exemple de code suivant montre comment utiliser`enable-directory-data-access`.

AWS CLI

Pour activer l'API Directory Service Data pour un annuaire

L'`enable-directory-data-access`exemple suivant active l'API Directory Service Data pour le répertoire spécifié.

```
aws ds enable-directory-data-access \
    --directory-id d-1234567890
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation ou désactivation de la gestion des utilisateurs et des groupes ou des données des services d' AWS annuaire dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous EnableDirectoryDataAccessà la section Référence des AWS CLI commandes.

`list-group-members`

L'exemple de code suivant montre comment utiliser`list-group-members`.

AWS CLI

Pour répertorier les membres du groupe d'un annuaire

L'`list-group-members`exemple suivant répertorie les membres du groupe spécifié dans le répertoire spécifié.

```
aws ds-data list-group-members \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

Sortie :

```
{
    "Members": [
```

```
        {
            "MemberType": "USER",
            "SAMAccountName": "Jane Doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        },
        {
            "MemberType": "USER",
            "SAMAccountName": "John Doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4569"
        }
    ],
    "DirectoryId": "d-1234567890",
    "MemberRealm": "corp.example.com",
    "Realm": "corp.example.com"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour des informations relatives à un groupe Microsoft AD AWS géré dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous ListGroupMembersà la section Référence des AWS CLI commandes.

`list-groups-for-member`

L'exemple de code suivant montre comment utiliser`list-groups-for-member`.

AWS CLI

Pour répertorier les membres du groupe d'un annuaire

L'`list-groups-for-member`exemple suivant répertorie l'appartenance à un groupe pour l'utilisateur spécifié dans le répertoire spécifié.

```
aws ds-data list-groups-for-member \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

Sortie :

```
{
    "Groups": [
        {
            "GroupScope": "Global",
```

```
            "GroupType": "Security",
            "SAMAccountName": "Domain Users",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "MemberRealm": "corp.example.com",
    "Realm": "corp.example.com"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous ListGroupsForMemberà la section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour répertorier les groupes d'un annuaire

L'`list-groups`exemple suivant répertorie les groupes du répertoire spécifié.

```
aws ds-data list-groups \
    --directory-id d-1234567890
```

Sortie :

```
{
    "Groups": [
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Administrators",
            "SID": "S-1-2-33-441"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Users",
```

```
            "SID": "S-1-2-33-442"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Guests",
            "SID": "S-1-2-33-443"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Print Operators",
            "SID": "S-1-2-33-444"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Backup Operators",
            "SID": "S-1-2-33-445"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Replicator",
            "SID": "S-1-2-33-446"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Remote Desktop Users",
            "SID": "S-1-2-33-447"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Network Configuration Operators",
            "SID": "S-1-2-33-448"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Performance Monitor Users",
            "SID": "S-1-2-33-449"
        },
```

```
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Performance Log Users",
            "SID": "S-1-2-33-450"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Distributed COM Users",
            "SID": "S-1-2-33-451"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "IIS_IUSRS",
            "SID": "S-1-2-33-452"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Cryptographic Operators",
            "SID": "S-1-2-33-453"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Event Log Readers",
            "SID": "S-1-2-33-454"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Certificate Service DCOM Access",
            "SID": "S-1-2-33-456"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Remote Access Servers",
            "SID": "S-1-2-33-457"
        },
        {
            "GroupScope": "BuiltinLocal",
```

```
            "GroupType": "Security",
            "SAMAccountName": "RDS Endpoint Servers",
            "SID": "S-1-2-33-458"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Management Servers",
            "SID": "S-1-2-33-459"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Hyper-V Administrators",
            "SID": "S-1-2-33-460"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Access Control Assistance Operators",
            "SID": "S-1-2-33-461"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Remote Management Users",
            "SID": "S-1-2-33-462"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Storage Replica Administrators",
            "SID": "S-1-2-33-463"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Computers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-789"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Controllers",
```

```
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-790"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Schema Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-791"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Enterprise Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-792"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Cert Publishers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-793"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-794"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Users",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-795"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Guests",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-796"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Group Policy Creator Owners",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-797"
        },
```

```
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "RAS and IAS Servers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-798"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Server Operators",
            "SID": "S-1-2-33-464"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Account Operators",
            "SID": "S-1-2-33-465"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Pre-Windows 2000 Compatible Access",
            "SID": "S-1-2-33-466"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Incoming Forest Trust Builders",
            "SID": "S-1-2-33-467"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Windows Authorization Access Group",
            "SID": "S-1-2-33-468"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Terminal Server License Servers",
            "SID": "S-1-2-33-469"
        },
        {
            "GroupScope": "DomainLocal",
```

```
            "GroupType": "Security",
            "SAMAccountName": "Allowed RODC Password Replication Group",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-798"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Denied RODC Password Replication Group",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-799"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Read-only Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-800"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Enterprise Read-only Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-801"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Cloneable Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-802"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Protected Users",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-803"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Key Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-804"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Enterprise Key Admins",
```

```
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-805"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "DnsAdmins",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "DnsUpdateProxy",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Admins",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4569"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWSAdministrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4570"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Object Management Service Accounts",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4571"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Private CA Connector for AD Delegated Group",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4572"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Application and Service Delegated Group",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4573"
        },
```

```
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4574"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated FSx Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4575"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Account Operators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4576"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Active Directory Based Activation
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4577"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Allowed to Authenticate Objects",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4578"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Allowed to Authenticate to Domain
Controllers",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4579"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Deleted Object Lifetime
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4580"
```

```
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Distributed File System
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4581"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Dynamic Host Configuration Protocol
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4582"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Enterprise Certificate Authority
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4583"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Fine Grained Password Policy
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4584"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Group Policy Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4585"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Managed Service Account
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4586"
        },
        {
            "GroupScope": "DomainLocal",
```

```
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Read Foreign Security Principals",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4587"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Remote Access Service Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4588"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Replicate Directory Changes
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4588"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Sites and Services Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4589"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated System Management Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4590"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Terminal Server Licensing
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4591"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated User Principal Name Suffix
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4592"
        },
        {
```

```
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Add Workstations To Domain Users",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4593"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Domain Name System Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4594"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Kerberos Delegation Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4595"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Server Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4596"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated MS-NPRC Non-Compliant Devices",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4597"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Remote Access",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4598"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Accounting",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4599"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Distribution",
```

```
            "SAMAccountName": "sales",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour des informations relatives à un groupe Microsoft AD AWS géré dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous ListGroupsà la section Référence des AWS CLI commandes.

`list-users`

L'exemple de code suivant montre comment utiliser`list-users`.

AWS CLI

Pour répertorier les utilisateurs d'un annuaire

L'`list-users`exemple suivant répertorie les utilisateurs du répertoire spécifié.

```
aws ds-data list-users \
    --directory-id d-1234567890
```

Sortie :

```
{
    "Users": [
        {
            "Enabled": true,
            "SAMAccountName": "Administrator",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-345"
        },
        {
            "Enabled": false,
            "SAMAccountName": "Guest",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-345"
        },
        {
```

```
            "Enabled": false,
            "SAMAccountName": "krbtgt",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-346"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Admin",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-347"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Richard Roe",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-348"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Jane Doe",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-349"
        },
        {
            "Enabled": true,
            "SAMAccountName": "AWS_WGnzYlN6YyY",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        },
        {
            "Enabled": true,
            "SAMAccountName": "john.doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`remove-group-member`

L'exemple de code suivant montre comment utiliser`remove-group-member`.

AWS CLI

Pour supprimer un membre d'un groupe d'un annuaire

L'`remove-group-member`exemple suivant supprime le membre du groupe spécifié du groupe spécifié dans le répertoire spécifié.

```
aws ds-data remove-group-member \
    --directory-id d-1234567890 \
    --group-name 'sales' \
    --member-name 'john.doe'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout et suppression de membres Microsoft AD AWS gérés à des groupes et de groupes à des groupes dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous RemoveGroupMemberà la section Référence des AWS CLI commandes.

`reset-user-password`

L'exemple de code suivant montre comment utiliser`reset-user-password`.

AWS CLI

Pour réinitialiser le mot de passe d'un utilisateur dans un annuaire

L'`reset-user-password`exemple suivant réinitialise et active l'utilisateur spécifié dans le répertoire spécifié.

```
aws ds reset-user-password \
    --directory-id d-1234567890 \
    --user-name 'john.doe' \
    --new-password 'password'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Réinitialisation et activation du mot de passe d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous ResetUserPasswordà la section Référence des AWS CLI commandes.

`search-groups`

L'exemple de code suivant montre comment utiliser`search-groups`.

AWS CLI

Pour rechercher un groupe dans un annuaire

L'`search-groups`exemple suivant recherche le groupe spécifié dans le répertoire spécifié.

```
aws ds-data search-groups \
    --directory-id d-1234567890 \
    --search-attributes 'SamAccountName' \
    --search-string 'sales'
```

Sortie :

```
{
    "Groups": [
        {
            "GroupScope": "Global",
            "GroupType": "Distribution",
            "SAMAccountName": "sales",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour des informations relatives à un groupe Microsoft AD AWS géré dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous SearchGroupsà la section Référence des AWS CLI commandes.

`search-users`

L'exemple de code suivant montre comment utiliser`search-users`.

AWS CLI

Pour rechercher un utilisateur dans un annuaire

L'`search-users`exemple suivant recherche l'utilisateur indiqué dans le répertoire indiqué.

```
aws ds-data search-users \
    --directory-id d-1234567890 \
    --search-attributes 'SamAccountName' \
    --Search-string 'john.doe'
```

Sortie :

```
{
    "Users": [
        {
            "Enabled": true,
            "SAMAccountName": "john.doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

Pour plus d'informations, consultez la section Affichage et mise à jour d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous SearchUsersà la section Référence des AWS CLI commandes.

`update-group`

L'exemple de code suivant montre comment utiliser`update-group`.

AWS CLI

Pour mettre à jour l'attribut d'un groupe dans un annuaire

L'`update-group`exemple suivant met à jour l'attribut spécifié pour le groupe spécifié dans le répertoire spécifié.

```
aws ds-data update-group \
```

```
    --directory-id d-1234567890 \
    --sam-account-name 'sales' \
    --update-type 'REPLACE' \
    --group-type 'Distribution'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Affichage et mise à jour des informations relatives à un groupe Microsoft AD AWS géré dans le Guide d'administration des services d'AWS annuaire.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupà la section Référence des AWS CLI commandes.

`update-user`

L'exemple de code suivant montre comment utiliser`update-user`.

AWS CLI

Pour mettre à jour l'attribut d'un utilisateur dans un annuaire

L'`update-user`exemple suivant met à jour l'attribut spécifié pour l'utilisateur spécifié dans le répertoire spécifié.

```
aws ds-data update-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe' \
    --update-type 'ADD' \
    --email-address 'example.corp.com'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Affichage et mise à jour d'un utilisateur Microsoft AD AWS géré dans le Guide d'administration du AWS Directory Service.

- Pour plus de détails sur l'API, reportez-vous UpdateUserà la section Référence des AWS CLI commandes.

AWS DMS exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS DMS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-resource`

L'exemple de code suivant montre comment utiliser`add-tags-to-resource`.

AWS CLI

Pour ajouter des balises à une ressource

L'`add-tags-to-resource`exemple suivant ajoute des balises à une instance de réplication.

```
aws dms add-tags-to-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
 \
    --tags Key=Environment,Value=PROD Key=Project,Value=dbMigration
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous AddTagsToResourceà la section Référence des AWS CLI commandes.

`create-endpoint`

L'exemple de code suivant montre comment utiliser`create-endpoint`.

AWS CLI

Pour créer un point de terminaison

L'`create-endpoint`exemple suivant crée un point de terminaison pour une source Amazon S3.

```
aws dms create-endpoint \
    --endpoint-type source \
    --engine-name s3 \
    --endpoint-identifier src-endpoint \
    --s3-settings file://s3-settings.json
```

Contenu de `s3-settings.json` :

```
{
    "BucketName":"my-corp-data",
    "BucketFolder":"sourcedata",
    "ServiceAccessRoleArn":"arn:aws:iam::123456789012:role/my-s3-access-role"
}
```

Sortie :

```
{
    "Endpoint": {
        "EndpointIdentifier": "src-endpoint",
        "EndpointType": "SOURCE",
        "EngineName": "s3",
        "EngineDisplayName": "Amazon S3",
        "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",
        "Status": "active",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U",
        "SslMode": "none",
        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
        "S3Settings": {
            "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
            "CsvRowDelimiter": "\\n",
            "CsvDelimiter": ",",
            "BucketFolder": "sourcedata",
            "BucketName": "my-corp-data",
            "CompressionType": "NONE",
            "EnableStatistics": true
        }
    }
```

```
}
```

Pour plus d'informations, consultez la section Utilisation des points de terminaison AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous CreateEndpointà la section Référence des AWS CLI commandes.

`create-event-subscription`

L'exemple de code suivant montre comment utiliser`create-event-subscription`.

AWS CLI

Pour répertorier les abonnements aux événements

L'`create-event-subscription`exemple suivant crée un abonnement à un événement sur une rubrique Amazon SNS ()my-sns-topic.

```
aws dms create-event-subscription \
    --subscription-name my-dms-events \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:my-sns-topic
```

Sortie :

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "my-dms-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
        "Status": "creating",
        "SubscriptionCreationTime": "2020-05-21 21:58:38.598",
        "Enabled": true
    }
}
```

Pour plus d'informations, consultez la section Utilisation des événements et des notifications dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, reportez-vous CreateEventSubscriptionà la section Référence des AWS CLI commandes.

`create-replication-instance`

L'exemple de code suivant montre comment utiliser`create-replication-instance`.

AWS CLI

Pour créer une instance de réplication

L'`create-replication-instance`exemple suivant crée une instance de réplication.

```
aws dms create-replication-instance \
    --replication-instance-identifier my-repl-instance \
    --replication-instance-class dms.t2.micro \
    --allocated-storage 5
```

Sortie :

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "creating",
        "AllocatedStorage": 5,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-f839b688",
                "Status": "active"
            }
        ],
        "ReplicationSubnetGroup": {
            "ReplicationSubnetGroupIdentifier": "default",
            "ReplicationSubnetGroupDescription": "default",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-42599426",
```

```
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-6746046b",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-cbfff283",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sat:12:35-sat:13:05",
        "PendingModifiedValues": {},
        "MultiAZ": false,
        "EngineVersion": "3.3.2",
        "AutoMinorVersionUpgrade": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:ZK2VQBUWFDBAWHIXHAYG5G2PKY",
```

```
        "PubliclyAccessible": true
    }
}
```

Pour plus d'informations, consultez la section Utilisation d'une instance de réplication AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous CreateReplicationInstanceà la section Référence des AWS CLI commandes.

`create-replication-subnet-group`

L'exemple de code suivant montre comment utiliser`create-replication-subnet-group`.

AWS CLI

Pour créer un groupe de sous-réseaux

L'`create-replication-subnet-group`exemple suivant crée un groupe composé de 3 sous-réseaux.

```
aws dms create-replication-subnet-group \
    --replication-subnet-group-identifier my-subnet-group \
    --replication-subnet-group-description "my subnet group" \
    --subnet-ids subnet-da327bf6 subnet-bac383e0 subnet-d7c825e8
```

Sortie :

```
{
    "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "my-subnet-group",
        "ReplicationSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-136a4c6a",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-da327bf6",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active"
```

```
            },
            {
                "SubnetIdentifier": "subnet-bac383e0",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-d7c825e8",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetStatus": "Active"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Configuration d'un réseau pour une instance de réplication dans le Guide AWS de l'utilisateur du Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous CreateReplicationSubnetGroupà la section Référence des AWS CLI commandes.

`create-replication-task`

L'exemple de code suivant montre comment utiliser`create-replication-task`.

AWS CLI

Pour créer une tâche de réplication

L'`create-replication-task`exemple suivant crée une tâche de réplication.

```
aws dms create-replication-task \
    --replication-task-identifier movedata \
    --source-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
    --target-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U \
    --replication-instance-arn $RI_ARN \
```

```
    --migration-type full-load \
    --table-mappings file://table-mappings.json
```

Contenu de `table-mappings.json` :

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "prodrep",
                "table-name": "%"
            },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

Sortie :

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted... ,
        "ReplicationTaskSettings": ...output omitted... ,
        "Status": "creating",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des tâches AWS DMS dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, reportez-vous CreateReplicationTaskà la section Référence des AWS CLI commandes.

`delete-connection`

L'exemple de code suivant montre comment utiliser`delete-connection`.

AWS CLI

Pour supprimer une connexion

L'`delete-connection`exemple suivant dissocie un point de terminaison d'une instance de réplication.

```
aws dms delete-connection \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

Sortie :

```
{
    "Connection": {
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "Status": "deleting",
        "EndpointIdentifier": "src-database-1",
        "ReplicationInstanceIdentifier": "my-repl-instance"
    }
}
```

Pour plus d'informations, consultez https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.creating.html dans le Guide de l'utilisateur du Database Migration AWS Service.

- Pour plus de détails sur l'API, reportez-vous DeleteConnectionà la section Référence des AWS CLI commandes.

`delete-endpoint`

L'exemple de code suivant montre comment utiliser`delete-endpoint`.

AWS CLI

Pour supprimer un point de terminaison

L'`delete-endpoint`exemple suivant supprime un point de terminaison.

```
aws dms delete-endpoint \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVXO4XZ4CYTSEG5XGMN2R3Y
```

Sortie :

```
{
    "Endpoint": {
        "EndpointIdentifier": "src-endpoint",
        "EndpointType": "SOURCE",
        "EngineName": "s3",
        "EngineDisplayName": "Amazon S3",
        "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",
        "Status": "deleting",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVXO4XZ4CYTSEG5XGMN2R3Y",
        "SslMode": "none",
        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
        "S3Settings": {
            "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
            "CsvRowDelimiter": "\\n",
            "CsvDelimiter": ",",
            "BucketFolder": "sourcedata",
            "BucketName": "my-corp-data",
            "CompressionType": "NONE",
            "EnableStatistics": true
        }
    }
}
```

Pour plus d'informations, consultez la section Utilisation des points de terminaison AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous DeleteEndpointà la section Référence des AWS CLI commandes.

`delete-event-subscription`

L'exemple de code suivant montre comment utiliser`delete-event-subscription`.

AWS CLI

Pour supprimer un abonnement à un événement

L'`delete-event-subscription`exemple suivant supprime un abonnement à une rubrique Amazon SNS.

```
aws dms delete-event-subscription \
    --subscription-name "my-dms-events"
```

Sortie :

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "my-dms-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
        "Status": "deleting",
        "SubscriptionCreationTime": "2020-05-21 21:58:38.598",
        "Enabled": true
    }
}
```

Pour plus d'informations, consultez la section Utilisation des événements et des notifications dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, reportez-vous DeleteEventSubscriptionà la section Référence des AWS CLI commandes.

`delete-replication-instance`

L'exemple de code suivant montre comment utiliser`delete-replication-instance`.

AWS CLI

Pour supprimer une instance de réplication

L'exemple de `delete-replication-instance` suivant supprime une instance de réplication.

```
aws dms delete-replication-instance \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

Sortie :

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "deleting",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-f839b688",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1e",
        "ReplicationSubnetGroup": {
            "ReplicationSubnetGroupIdentifier": "default",
            "ReplicationSubnetGroupDescription": "default",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-42599426",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
```

```
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-6746046b",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-cbfff283",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
        "PendingModifiedValues": {},
        "MultiAZ": true,
        "EngineVersion": "3.3.2",
        "AutoMinorVersionUpgrade": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "ReplicationInstancePublicIpAddress": "54.225.120.92",
        "ReplicationInstancePrivateIpAddress": "172.31.30.121",
```

```
        "ReplicationInstancePublicIpAddresses": [
            "54.225.120.92",
            "3.230.18.248"
        ],
        "ReplicationInstancePrivateIpAddresses": [
            "172.31.30.121",
            "172.31.75.90"
        ],
        "PubliclyAccessible": true,
        "SecondaryAvailabilityZone": "us-east-1b"
    }
}
```

Pour plus d'informations, consultez la section Utilisation d'une instance de réplication AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous DeleteReplicationInstanceà la section Référence des AWS CLI commandes.

`delete-replication-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-replication-subnet-group`.

AWS CLI

Pour supprimer un groupe de sous-réseaux

L'`delete-replication-subnet-group`exemple suivant supprime un groupe de sous-réseaux.

```
aws dms delete-replication-subnet-group \
--replication-subnet-group-identifier my-subnet-group
```

Sortie :

```
(none)
```

Pour plus d'informations, consultez la section Configuration d'un réseau pour une instance de réplication dans le Guide AWS de l'utilisateur du Database Migration Service.

- Pour plus de détails sur l'API, reportez-vous DeleteReplicationSubnetGroupà la section Référence des AWS CLI commandes.

`delete-replication-task`

L'exemple de code suivant montre comment utiliser`delete-replication-task`.

AWS CLI

Pour supprimer une tâche de réplication

L'`delete-replication-task`exemple suivant supprime une tâche de réplication.

```
aws dms delete-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Sortie :

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "deleting",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789988.677,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des tâches AWS DMS dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, reportez-vous DeleteReplicationTaskà la section Référence des AWS CLI commandes.

`describe-account-attributes`

L'exemple de code suivant montre comment utiliser`describe-account-attributes`.

AWS CLI

Pour décrire les attributs du compte

L'`describe-account-attributes`exemple suivant répertorie les attributs de votre AWS compte.

```
aws dms describe-account-attributes
```

Sortie :

```
{
    "AccountQuotas": [
        {
            "AccountQuotaName": "ReplicationInstances",
            "Used": 1,
            "Max": 20
        },
        {
            "AccountQuotaName": "AllocatedStorage",
            "Used": 5,
            "Max": 10000
        },

        ...remaining output omitted...

    ],
    "UniqueAccountIdentifier": "cqahfbfy5xee"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAccountAttributesà la section Référence des AWS CLI commandes.

`describe-certificates`

L'exemple de code suivant montre comment utiliser`describe-certificates`.

AWS CLI

Pour répertorier les certificats disponibles

L'`describe-certificates`exemple suivant répertorie les certificats disponibles dans votre AWS compte.

```
aws dms describe-certificates
```

Sortie :

```
{
    "Certificates": [
        {
            "CertificateIdentifier": "my-cert",
            "CertificateCreationDate": 1543259542.506,
            "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"

            ... remaining output omittted ...

        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du protocole SSL dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeCertificatesla section Référence des AWS CLI commandes.

`describe-connections`

L'exemple de code suivant montre comment utiliser`describe-connections`.

AWS CLI

Pour décrire les connexions

L'`describe-connections`exemple suivant répertorie les connexions que vous avez testées entre une instance de réplication et un point de terminaison.

```
aws dms describe-connections
```

Sortie :

```
{
    "Connections": [
        {
            "Status": "successful",
            "ReplicationInstanceIdentifier": "test",
            "EndpointArn": "arn:aws:dms:us-east-arn:aws:dms:us-
east-1:123456789012:endpoint:ZW5UAN6P4E77EC7YWHK4RZZ3BE",
            "EndpointIdentifier": "testsrc1",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:6UTDJGBOUS3VI3SUWA66XFJCJQ"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création de points de terminaison source et cible dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeConnectionsla section Référence des AWS CLI commandes.

`describe-endpoint-types`

L'exemple de code suivant montre comment utiliser`describe-endpoint-types`.

AWS CLI

Pour répertorier les types de terminaux disponibles

L'`describe-endpoint-types`exemple suivant répertorie les types de points de terminaison MySQL disponibles.

```
aws dms describe-endpoint-types \
    --filters "Name=engine-name,Values=mysql"
```

Sortie :

```
{
```

```
    "SupportedEndpointTypes": [
        {
            "EngineName": "mysql",
            "SupportsCDC": true,
            "EndpointType": "source",
            "EngineDisplayName": "MySQL"
        },
        {
            "EngineName": "mysql",
            "SupportsCDC": true,
            "EndpointType": "target",
            "EngineDisplayName": "MySQL"
        }
    ]
}
```

Pour plus d'informations, consultez la section Working with AWS DMS Endpoints < https:// docs.aws.amazon.com/dms/ latest/userguide/CHAP _Endpoints.html>`__ dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeEndpointTypesla section Référence des AWS CLI commandes.

`describe-endpoints`

L'exemple de code suivant montre comment utiliser`describe-endpoints`.

AWS CLI

Pour décrire les points de terminaison

L'`describe-endpoints`exemple suivant répertorie les points de terminaison de votre AWS compte.

```
aws dms describe-endpoints
```

Sortie :

```
{
    "Endpoints": [
        {
```

```
            "Username": "dms",
            "Status": "active",
            "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:SF2WOFLWYWKVEOHID2EKLP3SJI",
            "ServerName": "ec2-52-32-48-61.us-west-2.compute.amazonaws.com",
            "EndpointType": "SOURCE",
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/94d5c4e7-4e4c-44be-
b58a-c8da7adf57cd",
            "DatabaseName": "test",
            "EngineName": "mysql",
            "EndpointIdentifier": "pri100",
            "Port": 8193
        },
        {
            "Username": "admin",
            "Status": "active",
            "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:TJJZCIH3CJ24TJRU4VC32WEWFR",
            "ServerName": "test.example.com",
            "EndpointType": "SOURCE",
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/2431021b-1cf2-
a2d4-77b2-59a9e4bce323",
            "DatabaseName": "EMPL",
            "EngineName": "oracle",
            "EndpointIdentifier": "test",
            "Port": 1521
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des points de terminaison AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeEndpointsla section Référence des AWS CLI commandes.

`describe-event-categories`

L'exemple de code suivant montre comment utiliser`describe-event-categories`.

AWS CLI

Pour décrire les catégories d'événements

L'`describe-event-categories`exemple suivant répertorie les catégories d'événements disponibles.

```
aws dms describe-event-categories
```

Sortie :

```
{
    "EventCategoryGroupList": [
        {
            "SourceType": "replication-instance",
            "EventCategories": [
                "low storage",
                "configuration change",
                "maintenance",
                "deletion",
                "creation",
                "failover",
                "failure"
            ]
        },
        {
            "SourceType": "replication-task",
            "EventCategories": [
                "configuration change",
                "state change",
                "deletion",
                "creation",
                "failure"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des événements et des notifications dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir DescribeEventCategoriesla section Référence des AWS CLI commandes.

`describe-event-subscriptions`

L'exemple de code suivant montre comment utiliser`describe-event-subscriptions`.

AWS CLI

Pour décrire les abonnements aux événements

L'`describe-event-subscriptions`exemple suivant répertorie les abonnements aux événements à une rubrique Amazon SNS.

```
aws dms describe-event-subscriptions
```

Sortie :

```
{
    "EventSubscriptionsList": [
        {
            "CustomerAwsId": "123456789012",
            "CustSubscriptionId": "my-dms-events",
            "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
            "Status": "deleting",
            "SubscriptionCreationTime": "2020-05-21 22:28:51.924",
            "Enabled": true
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des événements et des notifications dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir DescribeEventSubscriptionsla section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour répertorier les événements DMS

L'`describe-events`exemple suivant répertorie les événements issus d'une instance de réplication.

```
aws dms describe-events \
    --source-type "replication-instance"
```

Sortie :

```
{
    "Events": [
        {
            "SourceIdentifier": "my-repl-instance",
            "SourceType": "replication-instance",
            "Message": "Replication application shutdown",
            "EventCategories": [],
            "Date": 1590771645.776
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des événements et des notifications dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir DescribeEventsla section Référence des AWS CLI commandes.

`describe-orderable-replication-instances`

L'exemple de code suivant montre comment utiliser`describe-orderable-replication-instances`.

AWS CLI

Pour décrire les instances de réplication commandables

L'`describe-orderable-replication-instances`exemple suivant répertorie les types d'instances de réplication que vous pouvez commander.

```
aws dms describe-orderable-replication-instances
```

Sortie :

```
{
    "OrderableReplicationInstances": [
        {
            "EngineVersion": "3.3.2",
            "ReplicationInstanceClass": "dms.c4.2xlarge",
            "StorageType": "gp2",
            "MinAllocatedStorage": 5,
            "MaxAllocatedStorage": 6144,
            "DefaultAllocatedStorage": 100,
            "IncludedAllocatedStorage": 100,
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
                "us-east-1f"
            ]
        },
        {
            "EngineVersion": "3.3.2",
            "ReplicationInstanceClass": "dms.c4.4xlarge",
            "StorageType": "gp2",
            "MinAllocatedStorage": 5,
            "MaxAllocatedStorage": 6144,
            "DefaultAllocatedStorage": 100,
            "IncludedAllocatedStorage": 100,
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
                "us-east-1f"
            ]
        },

        ...remaining output omitted...

    }
```

Pour plus d'informations, consultez la section Utilisation d'une instance de réplication AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeOrderableReplicationInstancesla section Référence des AWS CLI commandes.

`describe-refresh-schemas-status`

L'exemple de code suivant montre comment utiliser`describe-refresh-schemas-status`.

AWS CLI

Pour répertorier l'état d'actualisation d'un point de terminaison

L'`describe-refresh-schemas-status`exemple suivant renvoie le statut d'une demande d'actualisation précédente.

```
aws dms describe-refresh-schemas-status \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

Sortie :

```
{
    "RefreshSchemasStatus": {
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "Status": "successful",
        "LastRefreshDate": 1590786544.605
    }
}
```

- Pour plus de détails sur l'API, voir DescribeRefreshSchemasStatusla section Référence des AWS CLI commandes.

`describe-replication-instances`

L'exemple de code suivant montre comment utiliser`describe-replication-instances`.

AWS CLI

Pour décrire les instances de réplication

L'`describe-replication-instances`exemple suivant répertorie les instances de réplication de votre AWS compte.

```
aws dms describe-replication-instances
```

Sortie :

```
{
    "ReplicationInstances": [
        {
            "ReplicationInstanceIdentifier": "my-repl-instance",
            "ReplicationInstanceClass": "dms.t2.micro",
            "ReplicationInstanceStatus": "available",
            "AllocatedStorage": 5,
            "InstanceCreateTime": 1590011235.952,
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-f839b688",
                    "Status": "active"
                }
            ],
            "AvailabilityZone": "us-east-1e",
            "ReplicationSubnetGroup": {
                "ReplicationSubnetGroupIdentifier": "default",
                "ReplicationSubnetGroupDescription": "default",
                "VpcId": "vpc-136a4c6a",
                "SubnetGroupStatus": "Complete",
                "Subnets": [
                    {
                        "SubnetIdentifier": "subnet-da327bf6",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1a"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-42599426",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1d"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
```

```
                        "SubnetIdentifier": "subnet-bac383e0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1c"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-6746046b",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1f"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-d7c825e8",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1e"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-cbfff283",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    }
                ]
            },
            "PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
            "PendingModifiedValues": {
                "MultiAZ": true
            },
            "MultiAZ": false,
            "EngineVersion": "3.3.2",
            "AutoMinorVersionUpgrade": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
f7bc0f8e-1a3a-4ace-9faa-e8494fa3921a",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "ReplicationInstancePublicIpAddress": "3.230.18.248",
            "ReplicationInstancePrivateIpAddress": "172.31.75.90",
            "ReplicationInstancePublicIpAddresses": [
                "3.230.18.248"
```

```
            ],
            "ReplicationInstancePrivateIpAddresses": [
                "172.31.75.90"
            ],
            "PubliclyAccessible": true,
            "FreeUntil": 1590194829.267
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation d'une instance de réplication AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeReplicationInstancesla section Référence des AWS CLI commandes.

`describe-replication-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-replication-subnet-groups`.

AWS CLI

Pour afficher les groupes de sous-réseaux disponibles

L'`describe-replication-subnet-groups`exemple suivant répertorie les groupes de sous-réseaux disponibles.

```
aws dms describe-replication-subnet-groups \
    --filter "Name=replication-subnet-group-id,Values=my-subnet-group"
```

Sortie :

```
{
    "ReplicationSubnetGroups": [
        {
            "ReplicationSubnetGroupIdentifier": "my-subnet-group",
            "ReplicationSubnetGroupDescription": "my subnet group",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
```

```
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration d'un réseau pour une instance de réplication dans le Guide AWS de l'utilisateur du Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeReplicationSubnetGroupsla section Référence des AWS CLI commandes.

`describe-replication-task-assessment-results`

L'exemple de code suivant montre comment utiliser`describe-replication-task-assessment-results`.

AWS CLI

Pour répertorier les résultats des évaluations des tâches de réplication

L'`describe-replication-task-assessment-results`exemple suivant répertorie les résultats d'une évaluation de tâche précédente.

```
aws dms describe-replication-task-assessment-results
```

Sortie :

```
{
    "ReplicationTaskAssessmentResults": [
        {
            "ReplicationTaskIdentifier": "moveit2",
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
            "ReplicationTaskLastAssessmentDate": 1590790230.0,
            "AssessmentStatus": "No issues found",
            "AssessmentResultsFile": "moveit2/2020-05-29-22-10"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création d'un rapport d'évaluation des tâches dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir DescribeReplicationTaskAssessmentResultsla section Référence des AWS CLI commandes.

`describe-replication-tasks`

L'exemple de code suivant montre comment utiliser`describe-replication-tasks`.

AWS CLI

Pour décrire une tâche de réplication

L'`describe-replication-tasks`exemple suivant décrit les tâches de réplication en cours.

```
aws dms describe-replication-tasks
```

Sortie :

```
{
    "ReplicationTasks": [
        {
            "ReplicationTaskIdentifier": "moveit2",
```

```
            "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
            "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "MigrationType": "full-load",
            "TableMappings": ...output omitted... ,
            "ReplicationTaskSettings": ...output omitted... ,
            "Status": "stopped",
            "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
            "ReplicationTaskCreationDate": 1590524772.505,
            "ReplicationTaskStartDate": 1590619805.212,
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
            "ReplicationTaskStats": {
                "FullLoadProgressPercent": 100,
                "ElapsedTimeMillis": 0,
                "TablesLoaded": 0,
                "TablesLoading": 0,
                "TablesQueued": 0,
                "TablesErrored": 0,
                "FreshStartDate": 1590619811.528,
                "StartDate": 1590619811.528,
                "StopDate": 1590619842.068
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des tâches AWS DMS dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir DescribeReplicationTasksla section Référence des AWS CLI commandes.

`describe-schemas`

L'exemple de code suivant montre comment utiliser`describe-schemas`.

AWS CLI

Pour décrire les schémas de base de données

L'`describe-schemas`exemple suivant répertorie les tables disponibles sur un point de terminaison.

```
aws dms describe-schemas \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

Sortie :

```
{
    "Schemas": [
        "prodrep"
    ]
}
```

Pour plus d'informations, voir Ceci est le titre de la rubrique du Guide de l'utilisateur du Service AWS de Migration de Base de Données.

- Pour plus de détails sur l'API, voir DescribeSchemasla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises d'une instance de réplication.

```
aws dms list-tags-for-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "Project",
            "Value": "dbMigration"
```

```
        },
        {
            "Key": "Environment",
            "Value": "PROD"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`modify-endpoint`

L'exemple de code suivant montre comment utiliser`modify-endpoint`.

AWS CLI

Pour modifier un point de terminaison

L'`modify-endpoint`exemple suivant ajoute un attribut de connexion supplémentaire à un point de terminaison.

```
aws dms modify-endpoint \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U" \
    --extra-connection-attributes "compressionType=GZIP"
```

Sortie :

```
{
    "Endpoint": {
        "EndpointIdentifier": "src-endpoint",
        "EndpointType": "SOURCE",
        "EngineName": "s3",
        "EngineDisplayName": "Amazon S3",
        "ExtraConnectionAttributes":
 "compressionType=GZIP;csvDelimiter=,;csvRowDelimiter=\\n;",
        "Status": "active",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U",
```

```
        "SslMode": "none",
        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
        "S3Settings": {
            "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
            "CsvRowDelimiter": "\\n",
            "CsvDelimiter": ",",
            "BucketFolder": "",
            "BucketName": "",
            "CompressionType": "GZIP",
            "EnableStatistics": true
        }
    }
}
```

Pour plus d'informations, consultez la section Working with AWS DMS Endpoints < https://docs.aws.amazon.com/dms/ latest/userguide/CHAP _Endpoints.html>`__ dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir ModifyEndpointla section Référence des AWS CLI commandes.

`modify-event-subscription`

L'exemple de code suivant montre comment utiliser`modify-event-subscription`.

AWS CLI

Pour modifier l'abonnement à un événement

L'`modify-event-subscription`exemple suivant modifie le type de source d'un abonnement à un événement.

```
aws dms modify-event-subscription \
    --subscription-name "my-dms-events" \
    --source-type replication-task
```

Sortie :

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
```

```
        "CustSubscriptionId": "my-dms-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
        "Status": "modifying",
        "SubscriptionCreationTime": "2020-05-29 17:04:40.262",
        "SourceType": "replication-task",
        "Enabled": true
    }
}
```

Pour plus d'informations, consultez la section Utilisation des événements et des notifications dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir ModifyEventSubscriptionla section Référence des AWS CLI commandes.

`modify-replication-instance`

L'exemple de code suivant montre comment utiliser`modify-replication-instance`.

AWS CLI

Pour modifier une instance de réplication

L'`modify-replication-instance`exemple suivant modifie une instance de réplication afin qu'elle utilise un déploiement multi-AZ.

```
aws dms modify-replication-instance \
     --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
     --multi-az
```

Sortie :

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "available",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,

        ...output omitted...
```

```
        "PendingModifiedValues": {
            "MultiAZ": true
        },
        "MultiAZ": false,
        "EngineVersion": "3.3.2",
        "AutoMinorVersionUpgrade": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",

        ...output omitted...

    }
}
```

Pour plus d'informations, consultez la section Utilisation d'une instance de réplication AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir ModifyReplicationInstancela section Référence des AWS CLI commandes.

`modify-replication-subnet-group`

L'exemple de code suivant montre comment utiliser`modify-replication-subnet-group`.

AWS CLI

Pour modifier un groupe de sous-réseaux

L'`modify-replication-subnet-group`exemple suivant modifie les listes de sous-réseaux associées à un groupe de sous-réseaux.

```
aws dms modify-replication-subnet-group \
    --replication-subnet-group-identifier my-subnet-group \
    --subnet-id subnet-da327bf6 subnet-bac383e0
```

Sortie :

```
{
    "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "my-subnet-group",
        "ReplicationSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-136a4c6a",
        "SubnetGroupStatus": "Complete",
```

```
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-da327bf6",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-bac383e0",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Configuration d'un réseau pour une instance de réplication dans le Guide AWS de l'utilisateur du Database Migration Service.

- Pour plus de détails sur l'API, voir ModifyReplicationSubnetGroupla section Référence des AWS CLI commandes.

`modify-replication-task`

L'exemple de code suivant montre comment utiliser`modify-replication-task`.

AWS CLI

Pour modifier une tâche de réplication

L'`modify-replication-task`exemple suivant modifie les mappages de tables pour une tâche.

```
aws dms modify-replication-task \
    --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
    --table-mappings file://table-mappings.json
```

Contenu de `table-mappings.json` :

```
{
```

```
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "prodrep",
                "table-name": "ACCT_%"
            },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

Sortie :

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "modifying",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789424.653,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des tâches AWS DMS dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir ModifyReplicationTaskla section Référence des AWS CLI commandes.

`reboot-replication-instance`

L'exemple de code suivant montre comment utiliser`reboot-replication-instance`.

AWS CLI

Pour redémarrer une instance de réplication

L'exemple de `reboot-replication-instance` suivant redémarre une instance de réplication.

```
aws dms reboot-replication-instance \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

Sortie :

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "rebooting",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,
    ... output omitted ...
    }
}
```

Pour plus d'informations, consultez la section Utilisation d'une instance de réplication AWS DMS dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir RebootReplicationInstancela section Référence des AWS CLI commandes.

`refresh-schemas`

L'exemple de code suivant montre comment utiliser`refresh-schemas`.

AWS CLI

Pour actualiser les schémas de base de données

L'`refresh-schemas`exemple suivant demande à AWS DMS d'actualiser la liste des schémas d'un point de terminaison.

```
aws dms refresh-schemas \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

Sortie :

```
{
    "RefreshSchemasStatus": {
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "Status": "refreshing",
        "LastRefreshDate": 1590019949.103
    }
}
```

- Pour plus de détails sur l'API, voir RefreshSchemasla section Référence des AWS CLI commandes.

`reload-tables`

L'exemple de code suivant montre comment utiliser`reload-tables`.

AWS CLI

Pour actualiser la liste des tables disponibles sur un point de terminaison

L'`reload-tables`exemple suivant recharge la liste des tables disponibles sur un point de terminaison.

```
aws dms reload-tables \
    --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
    --tables-to-reload "SchemaName=prodrep,TableName=ACCT_BAL"
```

Sortie :

```
{
```

```
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- Pour plus de détails sur l'API, voir ReloadTablesla section Référence des AWS CLI commandes.

`remove-tags-from-resource`

L'exemple de code suivant montre comment utiliser`remove-tags-from-resource`.

AWS CLI

Pour supprimer des balises d'une instance de réplication

L'`remove-tags-from-resource`exemple suivant supprime les balises d'une instance de réplication.

```
aws dms remove-tags-from-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
 \
    --tag-keys Environment Project
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir RemoveTagsFromResourcela section Référence des AWS CLI commandes.

`start-replication-task-assessment`

L'exemple de code suivant montre comment utiliser`start-replication-task-assessment`.

AWS CLI

Pour démarrer une évaluation des tâches

L'`start-replication-task-assessment`exemple suivant lance une évaluation des tâches de réplication.

```
aws dms start-replication-task-assessment \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Sortie :

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "testing",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789988.677,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

Pour plus d'informations, consultez la section Création d'un rapport d'évaluation des tâches dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir StartReplicationTaskAssessmentla section Référence des AWS CLI commandes.

`start-replication-task`

L'exemple de code suivant montre comment utiliser`start-replication-task`.

AWS CLI

Pour démarrer une tâche de réplication

L'`command-name`exemple suivant répertorie les widgets disponibles dans votre AWS compte.

```
aws dms start-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \
    --start-replication-task-type reload-target
```

Sortie :

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted... ,
        "ReplicationTaskSettings": ...output omitted... ,
        "Status": "starting",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590619805.212,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des tâches AWS DMS dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir StartReplicationTaskla section Référence des AWS CLI commandes.

`stop-replication-task`

L'exemple de code suivant montre comment utiliser`stop-replication-task`.

AWS CLI

Pour arrêter une tâche

L'`stop-replication-task`exemple suivant arrête une tâche.

```
aws dms stop-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Sortie :

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "stopping",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789424.653,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des tâches AWS DMS dans le Guide de l'utilisateur du Service AWS de migration de base de données.

- Pour plus de détails sur l'API, voir StopReplicationTaskla section Référence des AWS CLI commandes.

`test-connection`

L'exemple de code suivant montre comment utiliser`test-connection`.

AWS CLI

Pour tester une connexion à un point de terminaison

L'`test-connection`exemple suivant teste si un point de terminaison est accessible depuis une instance de réplication.

```
aws dms test-connection \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

Sortie :

```
{
    "Connection": {
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "Status": "testing",
        "EndpointIdentifier": "src-database-1",
        "ReplicationInstanceIdentifier": "my-repl-instance"
    }
}
```

Pour plus d'informations, consultez la section Création de points de terminaison source et cible dans le Guide de l'utilisateur du AWS Database Migration Service.

- Pour plus de détails sur l'API, voir TestConnectionla section Référence des AWS CLI commandes.

Exemples d'Amazon DocumentDB utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon DocumentDB.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-resource`

L'exemple de code suivant montre comment utiliser`add-tags-to-resource`.

AWS CLI

Pour ajouter une ou plusieurs balises à une ressource spécifiée

L'`add-tags-to-resource`exemple suivant ajoute trois balises à`sample-cluster`. Une balise (`CropB`) possède un nom de clé mais aucune valeur.

```
aws docdb add-tags-to-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
    --tags Key="CropA",Value="Apple" Key="CropB" Key="CropC",Value="Corn"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous AddTagsToResourceà la section Référence des AWS CLI commandes.

`apply-pending-maintenance-action`

L'exemple de code suivant montre comment utiliser`apply-pending-maintenance-action`.

AWS CLI

Pour que les actions de maintenance en attente aient lieu au cours de la fenêtre de maintenance suivante

L'`apply-pending-maintenance-action`exemple suivant entraîne l'exécution de toutes les actions de mise à jour du système lors de la prochaine fenêtre de maintenance planifiée.

```
aws docdb apply-pending-maintenance-action \
--resource-identifier arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
--apply-action system-update \
```

```
--opt-in-type next-maintenance
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Appliquer les mises à jour d'Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ApplyPendingMaintenanceActionà la section Référence des AWS CLI commandes.

`copy-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`copy-db-cluster-parameter-group`.

AWS CLI

Pour dupliquer un groupe de paramètres de cluster de base de données existant

L'`copy-db-cluster-parameter-group`exemple suivant crée une copie du groupe de paramètres `custom-docdb3-6` nommé`custom-docdb3-6-copy`. Lors de la copie, il ajoute des balises au nouveau groupe de paramètres.

```
aws docdb copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier custom-docdb3-6 \
    --target-db-cluster-parameter-group-identifier custom-docdb3-6-copy \
    --target-db-cluster-parameter-group-description "Copy of custom-docdb3-6" \
    --tags Key="CopyNumber",Value="1" Key="Modifiable",Value="Yes"
```

Sortie :

```
{
    "DBClusterParameterGroup": {
        "DBParameterGroupFamily": "docdb3.6",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:12345678901:cluster-
pg:custom-docdb3-6-copy",
        "DBClusterParameterGroupName": "custom-docdb3-6-copy",
        "Description": "Copy of custom-docdb3-6"
    }
}
```

Pour plus d'informations, consultez Copier un groupe de paramètres de cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous CopyDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`copy-db-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`copy-db-cluster-snapshot`.

AWS CLI

Pour créer une copie d'un instantané

L'exemple `copy-db-cluster-snapshot` suivant fait une copie d'un `sample-cluster-snapshot` nommée `sample-cluster-snapshot-copy`. La copie contient toutes les balises de l'original plus une nouvelle balise avec le nom de la clé`CopyNumber`.

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --copy-tags \
    --tags Key="CopyNumber",Value="1"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Copier un instantané de cluster dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CopyDbClusterSnapshotà la section Référence des AWS CLI commandes.

`create-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`create-db-cluster-parameter-group`.

AWS CLI

Pour créer un groupe de paramètres de cluster Amazon DocumentDB

L'`create-db-cluster-parameter-group`exemple suivant crée le groupe de paramètres du cluster de base de données `sample-parameter-group` à l'aide de la `docdb3.6` famille.

```
aws docdb create-db-cluster-parameter-group \
```

```
    --db-cluster-parameter-group-name sample-parameter-group \
    --db-parameter-group-family docdb3.6 \
    --description "Sample parameter group based on docdb3.6"
```

Sortie :

```
{
    "DBClusterParameterGroup": {
        "Description": "Sample parameter group based on docdb3.6",
        "DBParameterGroupFamily": "docdb3.6",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-west-2:123456789012:cluster-
pg:sample-parameter-group",
        "DBClusterParameterGroupName": "sample-parameter-group"
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de paramètres de cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous CreateDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`create-db-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`create-db-cluster-snapshot`.

AWS CLI

Pour créer un instantané manuel du cluster Amazon DocumentDB

L'`create-db-cluster-snapshot`exemple suivant crée un instantané de cluster Amazon DB nommé sample-cluster-snapshot.

```
aws docdb create-db-cluster-snapshot \
   --db-cluster-identifier sample-cluster \
   --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Sortie :

```
{
    "DBClusterSnapshot": {
        "MasterUsername": "master-user",
```

```
        "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c",
            "us-west-2d",
            "us-west-2e",
            "us-west-2f"
        ],
        "SnapshotType": "manual",
        "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
        "EngineVersion": "3.6.0",
        "PercentProgress": 0,
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "Engine": "docdb",
        "DBClusterIdentifier": "sample-cluster",
        "Status": "creating",
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "Port": 0,
        "StorageEncrypted": false,
        "VpcId": "vpc-91280df6"
    }
}
```

Pour plus d'informations, consultez la section Création d'un instantané de cluster manuel dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous CreateDbClusterSnapshotà la section Référence des AWS CLI commandes.

`create-db-cluster`

L'exemple de code suivant montre comment utiliser`create-db-cluster`.

AWS CLI

Pour créer un cluster Amazon DocumentDB

L'`create-db-cluster`exemple suivant crée un cluster Amazon DocumentDB nommé `sample-cluster` avec la fenêtre de maintenance préférée le dimanche entre 20h30 et 11h00.

```
aws docdb create-db-cluster \
```

```
    --db-cluster-identifier sample-cluster \
    --engine docdb \
    --master-username master-user \
    --master-user-password password \
    --preferred-maintenance-window Sun:20:30-Sun:21:00
```

Sortie :

```
{
    "DBCluster": {
        "DBClusterParameterGroup": "default.docdb3.6",
        "AssociatedRoles": [],
        "DBSubnetGroup": "default",
        "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
        "Status": "creating",
        "Port": 27017,
        "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DBClusterIdentifier": "sample-cluster",
        "PreferredBackupWindow": "10:12-10:42",
        "AvailabilityZones": [
            "us-west-2d",
            "us-west-2f",
            "us-west-2e"
        ],
        "MasterUsername": "master-user",
        "BackupRetentionPeriod": 1,
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "StorageEncrypted": false,
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
        "MultiAZ": false,
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
```

```
        "EngineVersion": "3.6.0"
    }
}
```

Pour plus d'informations, consultez la section Création d'un cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous CreateDbClusterà la section Référence des AWS CLI commandes.

`create-db-instance`

L'exemple de code suivant montre comment utiliser`create-db-instance`.

AWS CLI

Pour créer une instance de cluster Amazon DocumentDB

L'`create-db-instance`exemple de code suivant crée l'instance `sample-cluster-instance-2` dans le cluster Amazon DocumentDB. `sample-cluster`

```
aws docdb create-db-instance \
    --db-cluster-identifier sample-cluster \
    --db-instance-class db.r4.xlarge \
    --db-instance-identifier sample-cluster-instance-2 \
    --engine docdb
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceStatus": "creating",
        "PendingModifiedValues": {
            "PendingCloudwatchLogsExports": {
                "LogTypesToEnable": [
                    "audit"
                ]
            }
        },
        "PubliclyAccessible": false,
        "PreferredBackupWindow": "00:00-00:30",
        "PromotionTier": 1,
        "EngineVersion": "3.6.0",
```

```
        "BackupRetentionPeriod": 3,
        "DBInstanceIdentifier": "sample-cluster-instance-2",
        "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
        "StorageEncrypted": false,
        "Engine": "docdb",
        "DBClusterIdentifier": "sample-cluster",
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-991cb8d0"
                }
            ],
            "DBSubnetGroupDescription": "default",
            "SubnetGroupStatus": "Complete",
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupName": "default"
        },
        "DBInstanceClass": "db.r4.xlarge",
        "VpcSecurityGroups": [
```

```
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
        "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4HO3ITE"
    }
}
```

Pour plus d'informations, consultez la section Ajouter une instance Amazon DocumentDB à un cluster dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDbInstanceà la section Référence des AWS CLI commandes.

`create-db-subnet-group`

L'exemple de code suivant montre comment utiliser`create-db-subnet-group`.

AWS CLI

Pour créer un groupe de sous-réseaux Amazon DocumentDB

L'`create-db-subnet-group`exemple suivant crée un groupe de sous-réseaux Amazon DocumentDB nommé. `sample-subnet-group`

```
aws docdb create-db-subnet-group \
    --db-subnet-group-description "a sample subnet group" \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids "subnet-29ab1025" "subnet-991cb8d0" "subnet-53ab3636"
```

Sortie :

```
{
    "DBSubnetGroup": {
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupName": "sample-subnet-group",
        "DBSubnetGroupDescription": "a sample subnet group",
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
```

```
        "Subnets": [
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-53ab3636",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                }
            },
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-991cb8d0",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                }
            },
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-29ab1025",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2c"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de sous-réseaux Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous CreateDbSubnetGroupà la section Référence des AWS CLI commandes.

`delete-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-db-cluster-parameter-group`.

AWS CLI

Pour supprimer un groupe de paramètres de cluster Amazon DocumentDB

L'`delete-db-cluster-parameter-group`exemple suivant supprime le groupe de paramètres Amazon DocumentDB. `sample-parameter-group`

```
aws docdb delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un groupe de paramètres de cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DeleteDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`delete-db-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`delete-db-cluster-snapshot`.

AWS CLI

Pour supprimer un instantané de cluster Amazon DocumentDB

L'`delete-db-cluster-snapshot`exemple suivant supprime l'instantané du cluster Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Sortie :

```
{
    "DBClusterSnapshot": {
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c",
            "us-west-2d"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "VpcId": "vpc-91280df6",
        "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
        "EngineVersion": "3.6.0",
        "Engine": "docdb",
```

```
        "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
        "Status": "available",
        "MasterUsername": "master-user",
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "PercentProgress": 100,
        "StorageEncrypted": false,
        "SnapshotType": "manual",
        "Port": 0
    }
}
```

Pour plus d'informations, consultez Supprimer un instantané de cluster dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDbClusterSnapshotà la section Référence des AWS CLI commandes.

`delete-db-cluster`

L'exemple de code suivant montre comment utiliser`delete-db-cluster`.

AWS CLI

Pour supprimer un cluster Amazon DocumentDB

L'`delete-db-cluster`exemple suivant supprime le cluster Amazon `sample-cluster` DocumentDB. Aucune sauvegarde du cluster n'est effectuée avant sa suppression. REMARQUE : Vous devez supprimer toutes les instances associées au cluster avant de pouvoir le supprimer.

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

Sortie :

```
{
    "DBCluster": {
        "DBClusterIdentifier": "sample-cluster",
        "DBSubnetGroup": "default",
        "EngineVersion": "3.6.0",
        "Engine": "docdb",
        "LatestRestorableTime": "2019-03-18T18:07:24.610Z",
```

```
        "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
        "StorageEncrypted": false,
        "EarliestRestorableTime": "2019-03-18T18:07:24.610Z",
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "MultiAZ": false,
        "MasterUsername": "master-user",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "Status": "available",
        "PreferredBackupWindow": "10:12-10:42",
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
        ],
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
        "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
        "AssociatedRoles": [],
        "DBClusterParameterGroup": "default.docdb3.6",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "BackupRetentionPeriod": 1,
        "DBClusterMembers": []
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DeleteDbClusterà la section Référence des AWS CLI commandes.

`delete-db-instance`

L'exemple de code suivant montre comment utiliser`delete-db-instance`.

AWS CLI

Pour supprimer une instance Amazon DocumentDB

L'`delete-db-instance`exemple suivant supprime l'instance Amazon `sample-cluster-instance-2` DocumentDB.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-cluster-instance-2
```

Sortie :

```
{
    "DBInstance": {
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-991cb8d0"
```

```
                }
            ],
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-91280df6",
            "SubnetGroupStatus": "Complete"
        },
        "PreferredBackupWindow": "00:00-00:30",
        "InstanceCreateTime": "2019-03-18T18:37:33.709Z",
        "DBInstanceClass": "db.r4.xlarge",
        "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4HO3ITE",
        "BackupRetentionPeriod": 3,
        "Engine": "docdb",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "AutoMinorVersionUpgrade": true,
        "PromotionTier": 1,
        "EngineVersion": "3.6.0",
        "Endpoint": {
            "Address": "sample-cluster-instance-2.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "DBInstanceIdentifier": "sample-cluster-instance-2",
        "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "PendingModifiedValues": {},
        "DBInstanceStatus": "deleting",
        "PubliclyAccessible": false,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZone": "us-west-2c",
        "StorageEncrypted": false
    }
}
```

Pour plus d'informations, consultez la section Suppression d'une instance Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DeleteDbInstanceà la section Référence des AWS CLI commandes.

`delete-db-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-db-subnet-group`.

AWS CLI

Pour supprimer un groupe de sous-réseaux Amazon DocumentDB

L'`delete-db-subnet-group`exemple suivant supprime le groupe de sous-réseaux Amazon DocumentDB. `sample-subnet-group`

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un groupe de sous-réseaux Amazon DocumentDB dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDbSubnetGroupà la section Référence des AWS CLI commandes.

`describe-db-cluster-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-parameter-groups`.

AWS CLI

Pour consulter les détails d'un ou de plusieurs groupes de paramètres de cluster Amazon DocumentDB

L'`describe-db-cluster-parameter-groups`exemple suivant affiche les détails du groupe de paramètres du cluster Amazon DocumentDB. `custom3-6-param-grp`

```
aws docdb describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name custom3-6-param-grp
```

Sortie :

```
{
    "DBClusterParameterGroups": [
        {
            "DBParameterGroupFamily": "docdb3.6",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:custom3-6-param-grp",
            "Description": "Custom docdb3.6 parameter group",
            "DBClusterParameterGroupName": "custom3-6-param-grp"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des groupes de paramètres du cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterParameterGroupsà la section Référence des AWS CLI commandes.

`describe-db-cluster-parameters`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-parameters`.

AWS CLI

Pour consulter la liste détaillée des paramètres d'un groupe de paramètres de cluster Amazon DocumentDB.

L'`describe-db-cluster-parameters`exemple suivant répertorie les paramètres du groupe de paramètres Amazon DocumentDB custom3-6-param-grp.

```
aws docdb describe-db-cluster-parameters \
     --db-cluster-parameter-group-name custom3-6-param-grp
```

Sortie :

```
{
    "Parameters": [
        {
            "DataType": "string",
            "ParameterName": "audit_logs",
```

```
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "Source": "system",
            "ApplyType": "dynamic",
            "AllowedValues": "enabled,disabled",
            "Description": "Enables auditing on cluster.",
            "ParameterValue": "disabled"
        },
        {
            "DataType": "string",
            "ParameterName": "tls",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "Source": "system",
            "ApplyType": "static",
            "AllowedValues": "disabled,enabled",
            "Description": "Config to enable/disable TLS",
            "ParameterValue": "enabled"
        },
        {
            "DataType": "string",
            "ParameterName": "ttl_monitor",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "Source": "user",
            "ApplyType": "dynamic",
            "AllowedValues": "disabled,enabled",
            "Description": "Enables TTL Monitoring",
            "ParameterValue": "enabled"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des paramètres du cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterParametersà la section Référence des AWS CLI commandes.

`describe-db-cluster-snapshot-attributes`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-snapshot-attributes`.

AWS CLI

Pour répertorier les noms et valeurs des attributs d'un instantané Amazon DocumentDB

L'`describe-db-cluster-snapshot-attributes`exemple suivant répertorie les noms et les valeurs des attributs pour l'instantané Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb describe-db-cluster-snapshot-attributes \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Sortie :

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": []
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

Pour plus d'informations, consultez la section Describe DBCluster SnapshotAttributes dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterSnapshotAttributesà la section Référence des AWS CLI commandes.

`describe-db-cluster-snapshots`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-snapshots`.

AWS CLI

Pour décrire les instantanés Amazon DocumentDB

L'`describe-db-cluster-snapshots`exemple suivant affiche les détails de l'instantané Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Sortie :

```
{
    "DBClusterSnapshots": [
        {
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c",
                "us-west-2d"
            ],
            "Status": "available",
            "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
            "SnapshotCreateTime": "2019-03-15T20:41:26.515Z",
            "SnapshotType": "manual",
            "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
            "DBClusterIdentifier": "sample-cluster",
            "MasterUsername": "master-user",
            "StorageEncrypted": false,
            "VpcId": "vpc-91280df6",
            "EngineVersion": "3.6.0",
            "PercentProgress": 100,
            "Port": 0,
            "Engine": "docdb",
            "ClusterCreateTime": "2019-03-15T20:29:58.836Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Décrire les DBCluster instantanés dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterSnapshotsà la section Référence des AWS CLI commandes.

`describe-db-clusters`

L'exemple de code suivant montre comment utiliser`describe-db-clusters`.

AWS CLI

Pour obtenir des informations détaillées sur un ou plusieurs clusters Amazon DocumentDB.

L'`describe-db-clusters`exemple suivant affiche les détails du cluster Amazon DocumentDB. `sample-cluster` En omettant le `--db-cluster-identifier` paramètre, vous pouvez obtenir des informations sur un maximum de 100 clusters.

```
aws docdb describe-db-clusters
    --db-cluster-identifier sample-cluster
```

Sortie :

```
{
    "DBClusters": [
        {
            "DBClusterParameterGroup": "default.docdb3.6",
            "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "PreferredBackupWindow": "00:00-00:30",
            "DBClusterIdentifier": "sample-cluster",
            "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
            "LatestRestorableTime": "2019-03-18T20:28:03.239Z",
            "MasterUsername": "master-user",
            "DBClusterMembers": [
                {
                    "PromotionTier": 1,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "IsClusterWriter": false,
                    "DBInstanceIdentifier": "sample-cluster"
                },
                {
                    "PromotionTier": 1,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "IsClusterWriter": true,
                    "DBInstanceIdentifier": "sample-cluster2"
                }
            ],
            "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-77186e0d",
                    "Status": "active"
```

```
                }
            ],
            "Engine": "docdb",
            "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "DBSubnetGroup": "default",
            "MultiAZ": true,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2c",
                "us-west-2b"
            ],
            "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
            "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
            "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
            "BackupRetentionPeriod": 3,
            "HostedZoneId": "ZNKXH85TT8WVW",
            "StorageEncrypted": false,
            "EnabledCloudwatchLogsExports": [
                "audit"
            ],
            "AssociatedRoles": [],
            "EngineVersion": "3.6.0",
            "Port": 27017,
            "Status": "available"
        }
    ]
}
```

Pour plus d'informations, consultez la section Décrire les clusters Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClustersà la section Référence des AWS CLI commandes.

`describe-db-engine-versions`

L'exemple de code suivant montre comment utiliser`describe-db-engine-versions`.

AWS CLI

Pour répertorier les versions du moteur Amazon DocumentDB disponibles

L'`describe-db-engine-versions`exemple suivant répertorie toutes les versions disponibles du moteur Amazon DocumentDB.

```
aws docdb describe-db-engine-versions \
    --engine docdb
```

Sortie :

```
{
    "DBEngineVersions": [
        {
            "DBEngineVersionDescription": "DocDB version 1.0.200837",
            "DBParameterGroupFamily": "docdb3.6",
            "EngineVersion": "3.6.0",
            "ValidUpgradeTarget": [],
            "DBEngineDescription": "Amazon DocumentDB (with MongoDB compatibility)",
            "SupportsLogExportsToCloudwatchLogs": true,
            "Engine": "docdb",
            "ExportableLogTypes": [
                "audit"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Décrire DBEngine les versions dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDbEngineVersionsà la section Référence des AWS CLI commandes.

`describe-db-instances`

L'exemple de code suivant montre comment utiliser`describe-db-instances`.

AWS CLI

Pour obtenir des informations sur les instances Amazon DocumentDB provisionnées

L'`describe-db-instances`exemple suivant affiche des informations sur l'instance Amazon DocumentDB. `sample-cluster-instance` En omettant le `--db-instance-identifier` paramètre, vous obtenez des informations sur un maximum de 100 instances.

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Sortie :

```
{
    "DBInstances": [
        {
            "Endpoint": {
                "HostedZoneId": "ZNKXH85TT8WVW",
                "Address": "sample-cluster-instance.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
                "Port": 27017
            },
            "PreferredBackupWindow": "00:00-00:30",
            "DBInstanceStatus": "available",
            "DBInstanceClass": "db.r4.large",
            "EnabledCloudwatchLogsExports": [
                "audit"
            ],
            "DBInstanceIdentifier": "sample-cluster-instance",
            "DBSubnetGroup": {
                "Subnets": [
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-4e26d263",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2a"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-afc329f4",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2c"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-53ab3636",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2d"
                        }
```

```
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-991cb8d0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2b"
                        }
                    }
                ],
                "DBSubnetGroupName": "default",
                "SubnetGroupStatus": "Complete",
                "DBSubnetGroupDescription": "default",
                "VpcId": "vpc-91280df6"
            },
            "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
            "Engine": "docdb",
            "StorageEncrypted": false,
            "AutoMinorVersionUpgrade": true,
            "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance",
            "PreferredMaintenanceWindow": "tue:08:39-tue:09:09",
            "VpcSecurityGroups": [
                {
                    "Status": "active",
                    "VpcSecurityGroupId": "sg-77186e0d"
                }
            ],
            "DBClusterIdentifier": "sample-cluster",
            "PendingModifiedValues": {},
            "BackupRetentionPeriod": 3,
            "PubliclyAccessible": false,
            "EngineVersion": "3.6.0",
            "PromotionTier": 1,
            "AvailabilityZone": "us-west-2c",
            "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA"
        }
    ]
}
```


Pour plus d'informations, consultez la section Décrire les instances Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DescribeDbInstancesà la section Référence des AWS CLI commandes.

`describe-db-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-db-subnet-groups`.

AWS CLI

Pour récupérer la liste des descriptions de sous-réseaux Amazon DocumentDB

L'`describe-db-subnet-groups`exemple suivant décrit les détails du sous-réseau Amazon DocumentDB nommé. `default`

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default
```

Sortie :

```
{
    "DBSubnetGroups": [
        {
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:default",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-4e26d263",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-afc329f4",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-53ab3636",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    }
                },
```

```
                {
                    "SubnetIdentifier": "subnet-991cb8d0",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    }
                }
            ],
            "DBSubnetGroupName": "default",
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupDescription": "default"
        }
    ]
}
```

Pour plus d'informations, consultez la section Décrire les groupes de sous-réseaux dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDbSubnetGroupsà la section Référence des AWS CLI commandes.

`describe-engine-default-cluster-parameters`

L'exemple de code suivant montre comment utiliser`describe-engine-default-cluster-parameters`.

AWS CLI

Pour décrire le moteur par défaut et les informations sur les paramètres système pour Amazon DocumentDB

L'`describe-engine-default-cluster-parameters`exemple suivant affiche les détails du moteur par défaut et les informations sur les paramètres système pour le groupe de paramètres Amazon DocumentDB. `docdb3.6`

```
aws docdb describe-engine-default-cluster-parameters \
    --db-parameter-group-family docdb3.6
```

Sortie :

```
{
    "EngineDefaults": {
```

```
        "DBParameterGroupFamily": "docdb3.6",
        "Parameters": [
            {
                "ApplyType": "dynamic",
                "ParameterValue": "disabled",
                "Description": "Enables auditing on cluster.",
                "Source": "system",
                "DataType": "string",
                "MinimumEngineVersion": "3.6.0",
                "AllowedValues": "enabled,disabled",
                "ParameterName": "audit_logs",
                "IsModifiable": true
            },
            {
                "ApplyType": "static",
                "ParameterValue": "enabled",
                "Description": "Config to enable/disable TLS",
                "Source": "system",
                "DataType": "string",
                "MinimumEngineVersion": "3.6.0",
                "AllowedValues": "disabled,enabled",
                "ParameterName": "tls",
                "IsModifiable": true
            },
            {
                "ApplyType": "dynamic",
                "ParameterValue": "enabled",
                "Description": "Enables TTL Monitoring",
                "Source": "system",
                "DataType": "string",
                "MinimumEngineVersion": "3.6.0",
                "AllowedValues": "disabled,enabled",
                "ParameterName": "ttl_monitor",
                "IsModifiable": true
            }
        ]
    }
}
```


Pour plus d'informations, consultez DescribeEngineDefaultClusterParametersle manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEngineDefaultClusterParametersà la section Référence des AWS CLI commandes.

`describe-event-categories`

L'exemple de code suivant montre comment utiliser`describe-event-categories`.

AWS CLI

Pour décrire toutes les catégories d'événements Amazon DocumentDB

L'`describe-event-categories`exemple suivant répertorie toutes les catégories du type de source d'événement Amazon DocumentDB. `db-instance`

```
aws docdb describe-event-categories \
    --source-type db-cluster
```

Sortie :

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "failover",
                "maintenance",
                "notification",
                "failure"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des catégories d'événements dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEventCategoriesà la section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour répertorier les événements Amazon DocumentDB

L'`describe-events`exemple suivant répertorie tous les événements Amazon DocumentDB des dernières 24 heures (1 440 minutes).

```
aws docdb describe-events \
    --duration 1440
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "Events": [
        {
            "EventCategories": [
                "failover"
            ],
            "Message": "Started cross AZ failover to DB instance: sample-cluster",
            "Date": "2019-03-18T21:36:29.807Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
            "SourceIdentifier": "sample-cluster",
            "SourceType": "db-cluster"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-18T21:36:40.793Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster",
            "SourceIdentifier": "sample-cluster",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [],
            "Message": "A new writer was promoted. Restarting database as a
 reader.",
            "Date": "2019-03-18T21:36:43.873Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
```

```
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-18T21:36:51.257Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "failover"
            ],
            "Message": "Completed failover to DB instance: sample-cluster",
            "Date": "2019-03-18T21:36:53.462Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
            "SourceIdentifier": "sample-cluster",
            "SourceType": "db-cluster"
        },
        {
            "Date": "2019-03-19T16:51:48.847Z",
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Updated parameter audit_logs to enabled with apply method
 pending-reboot",
            "SourceIdentifier": "custom3-6-param-grp",
            "SourceType": "db-parameter-group"
        },
        {
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Applying modification to database instance class",
            "Date": "2019-03-19T17:55:20.095Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance shutdown",
            "Date": "2019-03-19T17:56:31.127Z",
```

```
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Finished applying modification to DB instance class",
            "Date": "2019-03-19T18:00:45.822Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-19T18:00:53.397Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance shutdown",
            "Date": "2019-03-19T18:23:36.045Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-19T18:23:46.209Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
```

```
        {
            "Date": "2019-03-19T18:39:05.822Z",
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Updated parameter ttl_monitor to enabled with apply method
 immediate",
            "SourceIdentifier": "custom3-6-param-grp",
            "SourceType": "db-parameter-group"
        },
        {
            "Date": "2019-03-19T18:39:48.067Z",
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Updated parameter audit_logs to disabled with apply method
 immediate",
            "SourceIdentifier": "custom3-6-param-grp",
            "SourceType": "db-parameter-group"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des événements Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DescribeEventsà la section Référence des AWS CLI commandes.

`describe-orderable-db-instance-options`

L'exemple de code suivant montre comment utiliser`describe-orderable-db-instance-options`.

AWS CLI

Pour trouver les options d'instance Amazon DocumentDB, vous pouvez commander

L'`describe-orderable-db-instance-options`exemple suivant répertorie toutes les options d'instance pour Amazon DocumentDB pour une région.

```
aws docdb describe-orderable-db-instance-options \
```

```
    --engine docdb \
    --region us-east-1
```

Sortie :

```
{
    "OrderableDBInstanceOptions": [
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.16xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
```

```
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.2xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.4xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
```

```
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.8xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.large",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.xlarge",
```

```
            "LicenseModel": "na",
            "Engine": "docdb"
        }
    ]
}
```

Pour plus d'informations, consultez la section Ajouter une instance Amazon DocumentDB à un cluster dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeOrderableDbInstanceOptionsà la section Référence des AWS CLI commandes.

`describe-pending-maintenance-actions`

L'exemple de code suivant montre comment utiliser`describe-pending-maintenance-actions`.

AWS CLI

Pour répertorier vos actions de maintenance Amazon DocumentDB en attente

L'`describe-pending-maintenance-actions`exemple suivant répertorie toutes vos actions de maintenance Amazon DocumentDB en attente.

```
aws docdb describe-pending-maintenance-actions
```

Sortie :

```
{
    "PendingMaintenanceActions": []
}
```

Pour plus d'informations, consultez la section Maintenance d'Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous DescribePendingMaintenanceActionsà la section Référence des AWS CLI commandes.

`failover-db-cluster`

L'exemple de code suivant montre comment utiliser`failover-db-cluster`.

AWS CLI

Pour forcer un cluster Amazon DocumentDB à basculer vers une réplique

L'`failover-db-cluster`exemple suivant provoque le basculement de l'instance principale du cluster d'échantillons Amazon DocumentDB vers une réplique.

```
aws docdb failover-db-cluster \
    --db-cluster-identifier sample-cluster
```

Sortie :

```
{
    "DBCluster": {
        "AssociatedRoles": [],
        "DBClusterIdentifier": "sample-cluster",
        "EngineVersion": "3.6.0",
        "DBSubnetGroup": "default",
        "MasterUsername": "master-user",
        "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
            "us-west-2b"
        ],
        "LatestRestorableTime": "2019-03-18T21:35:23.548Z",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "PreferredBackupWindow": "00:00-00:30",
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "StorageEncrypted": false,
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "MultiAZ": true,
        "Status": "available",
        "DBClusterMembers": [
            {
```

```
                "DBClusterParameterGroupStatus": "in-sync",
                "IsClusterWriter": false,
                "DBInstanceIdentifier": "sample-cluster",
                "PromotionTier": 1
            },
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "IsClusterWriter": true,
                "DBInstanceIdentifier": "sample-cluster2",
                "PromotionTier": 2
            }
        ],
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "DBClusterParameterGroup": "default.docdb3.6",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "BackupRetentionPeriod": 3,
        "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "Engine": "docdb"
    }
}
```

Pour plus d'informations, consultez Amazon DocumentDB Failover dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous FailoverDbClusterà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier toutes les balises d'une ressource Amazon DocumentDB

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises du cluster Amazon DocumentDB. `sample-cluster`

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "A",
            "Value": "ALPHA"
        },
        {
            "Key": "B",
            "Value": ""
        },
        {
            "Key": "C",
            "Value": "CHARLIE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les balises d'une ressource Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`modify-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`modify-db-cluster-parameter-group`.

AWS CLI

Pour modifier un groupe de paramètres de cluster de base de données Amazon DocumentDB

L'`modify-db-cluster-parameter-group`exemple suivant modifie le `custom3-6-param-grp` groupe de paramètres du cluster Amazon DocumentDB en définissant les deux `audit_logs` paramètres `ttl_monitor` et en les définissant sur Activé. Les modifications sont appliquées au prochain redémarrage.

```
aws docdb modify-db-cluster-parameter-group \
```

```
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --
parameters ParameterName=audit_logs,ParameterValue=enabled,ApplyMethod=pending-
reboot \

  ParameterName=ttl_monitor,ParameterValue=enabled,ApplyMethod=pending-reboot
```

Sortie :

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

Pour plus d'informations, consultez la section Modification d'un groupe de paramètres de cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`modify-db-cluster-snapshot-attribute`

L'exemple de code suivant montre comment utiliser`modify-db-cluster-snapshot-attribute`.

AWS CLI

Exemple 1 : pour ajouter un attribut à un instantané Amazon DocumentDB

L'`modify-db-cluster-snapshot-attribute`exemple suivant ajoute quatre valeurs d'attribut à un instantané de cluster Amazon DocumentDB.

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add 123456789011 123456789012 123456789013
```

Sortie :

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
```

```
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789011",
                    "123456789012",
                    "123456789013"
                ]
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

Exemple 2 : pour supprimer des attributs d'un instantané Amazon DocumentDB

L'`modify-db-cluster-snapshot-attribute`exemple suivant supprime deux valeurs d'attribut d'un instantané de cluster Amazon DocumentDB.

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove 123456789012
```

Sortie :

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789011",
                    "123456789013"
                ]
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

Pour plus d'informations, consultez Modifier DBCluster SnapshotAttribute dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterSnapshotAttributeà la section Référence des AWS CLI commandes.

`modify-db-cluster`

L'exemple de code suivant montre comment utiliser`modify-db-cluster`.

AWS CLI

Pour modifier un cluster Amazon DocumentDB

L'`modify-db-cluster`exemple suivant modifie le `sample-cluster` cluster Amazon DocumentDB en fixant à 7 jours la période de rétention pour les sauvegardes automatiques et en modifiant les fenêtres préférées pour les sauvegardes et la maintenance. Toutes les modifications sont appliquées lors de la fenêtre de maintenance suivante.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --no-apply-immediately \
    --backup-retention-period 7 \
    --preferred-backup-window 18:00-18:30 \
    --preferred-maintenance-window sun:20:00-sun:20:30
```

Sortie :

```
{
    "DBCluster": {
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DBClusterMembers": [
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster",
                "IsClusterWriter": true,
                "PromotionTier": 1
            },
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster2",
                "IsClusterWriter": false,
                "PromotionTier": 2
```

```
            }
        ],
        "HostedZoneId": "ZNKXH85TT8WVW",
        "StorageEncrypted": false,
        "PreferredBackupWindow": "18:00-18:30",
        "MultiAZ": true,
        "EngineVersion": "3.6.0",
        "MasterUsername": "master-user",
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DBSubnetGroup": "default",
        "LatestRestorableTime": "2019-03-18T22:08:13.408Z",
        "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
        "PreferredMaintenanceWindow": "sun:20:00-sun:20:30",
        "AssociatedRoles": [],
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "Engine": "docdb",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
            "us-west-2b"
        ],
        "Status": "available",
        "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "Port": 27017
    }
}
```


Pour plus d'informations, consultez la section Modification d'un cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterà la section Référence des AWS CLI commandes.

`modify-db-instance`

L'exemple de code suivant montre comment utiliser`modify-db-instance`.

AWS CLI

Pour modifier une instance Amazon DocumentDB

L'`modify-db-instance`exemple suivant modifie l'instance Amazon DocumentDB en remplaçant sa classe d'`sample-cluster2`instance par et son `db.r4.4xlarge` niveau de promotion par. `5` Les modifications sont appliquées immédiatement mais ne sont visibles qu'une fois que le statut des instances est disponible.

```
aws docdb modify-db-instance \
    --db-instance-identifier sample-cluster2 \
    --apply-immediately \
    --db-instance-class db.r4.4xlarge \
    --promotion-tier 5
```

Sortie :

```
{
    "DBInstance": {
        "EngineVersion": "3.6.0",
        "StorageEncrypted": false,
        "DBInstanceClass": "db.r4.large",
        "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
        "AutoMinorVersionUpgrade": true,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "PreferredBackupWindow": "18:00-18:30",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "AvailabilityZone": "us-west-2f",
```

```
        "DBInstanceIdentifier": "sample-cluster2",
        "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
        "Engine": "docdb",
        "BackupRetentionPeriod": 7,
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-4e26d263",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-afc329f4",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-53ab3636",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-991cb8d0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ],
            "VpcId": "vpc-91280df6"
        },
        "PromotionTier": 2,
        "Endpoint": {
            "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
```

```
            "Port": 27017
        },
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
        "DBClusterIdentifier": "sample-cluster",
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "PendingModifiedValues": {
            "DBInstanceClass": "db.r4.4xlarge"
        },
        "PubliclyAccessible": false,
        "DBInstanceStatus": "available"
    }
}
```

Pour plus d'informations, consultez la section Modification d'une instance Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ModifyDbInstanceà la section Référence des AWS CLI commandes.

`modify-db-subnet-group`

L'exemple de code suivant montre comment utiliser`modify-db-subnet-group`.

AWS CLI

Pour modifier un groupe de sous-réseaux Amazon DocumentDB

L'`modify-db-subnet-group`exemple suivant modifie le groupe de sous-réseaux `sample-subnet-group` en ajoutant les sous-réseaux spécifiés et une nouvelle description.

```
aws docdb modify-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids subnet-b3806e8f subnet-53ab3636 subnet-991cb8d0 \
    --db-subnet-group-description "New subnet description"
```

Sortie :

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "sample-subnet-group",
        "SubnetGroupStatus": "Complete",
```

```
        "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupDescription": "New subnet description",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-b3806e8f",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            },
            {
                "SubnetIdentifier": "subnet-53ab3636",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2c"
                }
            },
            {
                "SubnetIdentifier": "subnet-991cb8d0",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Modification d'un groupe de sous-réseaux Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ModifyDbSubnetGroupà la section Référence des AWS CLI commandes.

`reboot-db-instance`

L'exemple de code suivant montre comment utiliser`reboot-db-instance`.

AWS CLI

Pour redémarrer une instance Amazon DocumentDB

L'`reboot-db-instance`exemple suivant redémarre l'instance Amazon `sample-cluster2` DocumentDB.

```
aws docdb reboot-db-instance \
    --db-instance-identifier sample-cluster2
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "DBInstance": {
        "PreferredBackupWindow": "18:00-18:30",
        "DBInstanceIdentifier": "sample-cluster2",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBSubnetGroup": {
            "VpcId": "vpc-91280df6",
            "Subnets": [
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
```

```
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default"
        },
        "PendingModifiedValues": {},
        "Endpoint": {
            "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "StorageEncrypted": false,
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
        "AutoMinorVersionUpgrade": true,
        "Engine": "docdb",
        "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
        "EngineVersion": "3.6.0",
        "PromotionTier": 5,
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "sample-cluster",
        "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
        "PubliclyAccessible": false,
        "DBInstanceClass": "db.r4.4xlarge",
        "AvailabilityZone": "us-west-2d",
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "DBInstanceStatus": "rebooting"
    }
}
```


Pour plus d'informations, consultez la section Redémarrage d'une base de données Amazon ILnstance DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous RebootDbInstanceà la section Référence des AWS CLI commandes.

`remove-tags-from-resource`

L'exemple de code suivant montre comment utiliser`remove-tags-from-resource`.

AWS CLI

Pour supprimer des balises d'une ressource Amazon DocumentDB

L'`remove-tags-from-resource`exemple suivant supprime la balise dont la clé est nommée `B` dans le cluster Amazon DocumentDB. `sample-cluster`

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
    --tag-keys B
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer des balises d'un document Amazon DBResource dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RemoveTagsFromResourceà la section Référence des AWS CLI commandes.

`reset-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`reset-db-cluster-parameter-group`.

AWS CLI

Pour rétablir la valeur par défaut du paramètre spécifié dans un groupe de paramètres Amazon DocumentDB

L'`reset-db-cluster-parameter-group`exemple suivant rétablit la valeur par défaut du paramètre `ttl_monitor` du `custom3-6-param-grp` groupe de paramètres Amazon DocumentDB.

```
aws docdb reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters ParameterName=ttl_monitor,ApplyMethod=immediate
```

Sortie :

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

Pour plus d'informations, consultez le titre du manuel Amazon DocumentDB Developer Guide.

Pour rétablir les valeurs par défaut spécifiées ou toutes les valeurs des paramètres dans un groupe de paramètres Amazon DocumentDB

L'`reset-db-cluster-parameter-group`exemple suivant rétablit la valeur par défaut de tous les paramètres du `custom3-6-param-grp` groupe de paramètres Amazon DocumentDB.

```
aws docdb reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --reset-all-parameters
```

Sortie :

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

Pour plus d'informations, consultez la section Réinitialisation d'un groupe de paramètres de cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous ResetDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`restore-db-cluster-from-snapshot`

L'exemple de code suivant montre comment utiliser`restore-db-cluster-from-snapshot`.

AWS CLI

Pour restaurer un cluster Amazon DocumentDB à partir d'un instantané automatique ou manuel

L'`restore-db-cluster-from-snapshot`exemple suivant crée un nouveau cluster Amazon DocumentDB nommé `sample-cluster-2019-03-16-00-01-restored` à partir de l'instantané. `rds:sample-cluster-2019-03-16-00-01`

```
aws docdb restore-db-cluster-from-snapshot \
    --db-cluster-identifier sample-cluster-2019-03-16-00-01-restored \
    --engine docdb \
    --snapshot-identifier rds:sample-cluster-2019-03-16-00-01
```

Sortie :

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
            "us-west-2b"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
```

```
        ]
    }
}
```

Pour plus d'informations, consultez la section Restauration à partir d'un instantané de cluster dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RestoreDbClusterFromSnapshotà la section Référence des AWS CLI commandes.

`restore-db-cluster-to-point-in-time`

L'exemple de code suivant montre comment utiliser`restore-db-cluster-to-point-in-time`.

AWS CLI

Pour restaurer un cluster Amazon DocumentDB à point-in-time partir d'un instantané manuel

L'`restore-db-cluster-to-point-in-time`exemple suivant utilise le `sample-cluster-snapshot` pour créer un nouveau cluster Amazon DocumentDB`sample-cluster-pit`, en utilisant l'heure de restauration la plus récente.

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-pit \
    --source-db-cluster-identifier arn:aws:rds:us-
west-2:123456789012:cluster:sample-cluster \
    --use-latest-restorable-time
```

Sortie :

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "BackupRetentionPeriod": 3,
        "MasterUsername": "master-user",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "PreferredBackupWindow": "00:00-00:30",
        "MultiAZ": false,
        "DBClusterIdentifier": "sample-cluster-pit",
        "DBSubnetGroup": "default",
        "ClusterCreateTime": "2019-04-03T15:55:21.320Z",
        "AssociatedRoles": [],
```

```
        "DBClusterParameterGroup": "default.docdb3.6",
        "DBClusterMembers": [],
        "Status": "creating",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2d",
            "us-west-2b"
        ],
        "ReaderEndpoint": "sample-cluster-pit.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "Port": 27017,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "Endpoint": "sample-cluster-pit.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DbClusterResourceId": "cluster-NLCABBXOSE2QPQ4GOLZIFWEPLM",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
pit"
    }
}
```

Pour plus d'informations, consultez la section Restauration d'un instantané à un moment donné dans le manuel Amazon DocumentDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RestoreDbClusterToPointInTimeà la section Référence des AWS CLI commandes.

`start-db-cluster`

L'exemple de code suivant montre comment utiliser`start-db-cluster`.

AWS CLI

Pour démarrer un cluster Amazon DocumentDB arrêté

L'`start-db-cluster`exemple suivant démarre le cluster Amazon DocumentDB spécifié.

```
aws docdb start-db-cluster \
    --db-cluster-identifier sample-cluster
```

Sortie :

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
```

```
}
```

Pour plus d'informations, consultez la section Arrêter et démarrer un cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous StartDbClusterà la section Référence des AWS CLI commandes.

`stop-db-cluster`

L'exemple de code suivant montre comment utiliser`stop-db-cluster`.

AWS CLI

Pour arrêter un cluster Amazon DocumentDB en cours d'exécution

L'`stop-db-cluster`exemple suivant arrête le cluster Amazon DocumentDB spécifié.

```
aws docdb stop-db-cluster \
    --db-cluster-identifier sample-cluster
```

Sortie :

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
```

```
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Arrêter et démarrer un cluster Amazon DocumentDB dans le manuel du développeur Amazon DocumentDB.

- Pour plus de détails sur l'API, reportez-vous StopDbClusterà la section Référence des AWS CLI commandes.

Exemples DynamoDB utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de DynamoDB.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-get-item`

L'exemple de code suivant montre comment utiliser`batch-get-item`.

AWS CLI

Pour récupérer plusieurs éléments d'un tableau

L'`batch-get-item`exemple suivant lit plusieurs éléments de la `MusicCollection` table à l'aide d'un lot de trois `GetItem` demandes et demande le nombre d'unités de capacité de lecture consommées par l'opération. La commande renvoie uniquement l'`AlbumTitle`attribut.

```
aws dynamodb batch-get-item \
    --request-items file://request-items.json \
    --return-consumed-capacity TOTAL
```

Contenu de `request-items.json` :

```
{
    "MusicCollection": {
        "Keys": [
            {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Scared of My Shadow"}
            }
        ],
        "ProjectionExpression":"AlbumTitle"
    }
}
```

Sortie :

```
{
    "Responses": {
        "MusicCollection": [
            {
                "AlbumTitle": {
                    "S": "Somewhat Famous"
                }
            },
            {
                "AlbumTitle": {
                    "S": "Blue Sky Blues"
                }
            },
            {
                "AlbumTitle": {
                    "S": "Louder Than Ever"
                }
            }
        ]
    },
    "UnprocessedKeys": {},
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 1.5
        }
    ]
}
```

Pour plus d'informations, consultez Batch Operations dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchGetItemà la section Référence des AWS CLI commandes.

`batch-write-item`

L'exemple de code suivant montre comment utiliser`batch-write-item`.

AWS CLI

Pour ajouter plusieurs éléments à un tableau

L'`batch-write-item`exemple suivant ajoute trois nouveaux éléments à la `MusicCollection` table à l'aide d'un lot de trois `PutItem` demandes. Il demande également des informations sur le nombre d'unités de capacité d'écriture consommées par l'opération et sur les collections d'articles modifiées par l'opération.

```
aws dynamodb batch-write-item \
    --request-items file://request-items.json \
    --return-consumed-capacity INDEXES \
    --return-item-collection-metrics SIZE
```

Contenu de `request-items.json` :

```
{
    "MusicCollection": [
        {
            "PutRequest": {
                "Item": {
                    "Artist": {"S": "No One You Know"},
                    "SongTitle": {"S": "Call Me Today"},
                    "AlbumTitle": {"S": "Somewhat Famous"}
                }
            }
        },
        {
            "PutRequest": {
                "Item": {
                    "Artist": {"S": "Acme Band"},
                    "SongTitle": {"S": "Happy Day"},
                    "AlbumTitle": {"S": "Songs About Life"}
                }
            }
        },
        {
            "PutRequest": {
                "Item": {
                    "Artist": {"S": "No One You Know"},
                    "SongTitle": {"S": "Scared of My Shadow"},
                    "AlbumTitle": {"S": "Blue Sky Blues"}
                }
            }
        }
    ]
```

```
}
```

Sortie :

```
{
    "UnprocessedItems": {},
    "ItemCollectionMetrics": {
        "MusicCollection": [
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "No One You Know"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            },
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "Acme Band"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            }
        ]
    },
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 6.0,
            "Table": {
                "CapacityUnits": 3.0
            },
            "LocalSecondaryIndexes": {
                "AlbumTitleIndex": {
                    "CapacityUnits": 3.0
                }
```

```
            }
        }
    ]
}
```

Pour plus d'informations, consultez Batch Operations dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous BatchWriteItemà la section Référence des AWS CLI commandes.

`create-backup`

L'exemple de code suivant montre comment utiliser`create-backup`.

AWS CLI

Pour créer une sauvegarde pour une table DynamoDB existante

L'`create-backup`exemple suivant crée une sauvegarde de la `MusicCollection` table.

```
aws dynamodb create-backup \
    --table-name MusicCollection \
    --backup-name MusicCollectionBackup
```

Sortie :

```
{
    "BackupDetails": {
        "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a",
        "BackupName": "MusicCollectionBackup",
        "BackupSizeBytes": 0,
        "BackupStatus": "CREATING",
        "BackupType": "USER",
        "BackupCreationDateTime": 1576616366.715
    }
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous CreateBackupà la section Référence des AWS CLI commandes.

`create-global-table`

L'exemple de code suivant montre comment utiliser`create-global-table`.

AWS CLI

Pour créer une table globale

L'`create-global-table`exemple suivant crée une table globale à partir de deux tables identiques dans les AWS régions distinctes spécifiées.

```
aws dynamodb create-global-table \
    --global-table-name MusicCollection \
    --replication-group RegionName=us-east-2 RegionName=us-east-1 \
    --region us-east-2
```

Sortie :

```
{
    "GlobalTableDescription": {
        "ReplicationGroup": [
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
        "CreationDateTime": 1576625818.532,
        "GlobalTableStatus": "CREATING",
        "GlobalTableName": "MusicCollection"
    }
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous CreateGlobalTableà la section Référence des AWS CLI commandes.

`create-table`

L'exemple de code suivant montre comment utiliser`create-table`.

AWS CLI

Exemple 1 : pour créer un tableau avec des balises

L'`create-table`exemple suivant utilise les attributs et le schéma de clés spécifiés pour créer une table nomméeMusicCollection. Cette table utilise le débit provisionné et est chiffrée au repos à l'aide de la clé CMK AWS détenue par défaut. La commande applique également une balise à la table, avec une clé Owner et une valeur deblueTeam.

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
    --tags Key=Owner,Value=blueTeam
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
```

```
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "TableName": "MusicCollection",
        "TableStatus": "CREATING",
        "KeySchema": [
            {
                "KeyType": "HASH",
                "AttributeName": "Artist"
            },
            {
                "KeyType": "RANGE",
                "AttributeName": "SongTitle"
            }
        ],
        "ItemCount": 0,
        "CreationDateTime": "2020-05-26T16:04:41.627000-07:00",
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 2 : pour créer une table en mode à la demande

L'exemple suivant crée une table appelée `MusicCollection` en mode à la demande, plutôt qu'en mode débit provisionné. Cela est utile pour les tables dont les charges de travail sont imprévisibles.

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-27T11:44:10.807000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 0,
            "WriteCapacityUnits": 0
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "BillingModeSummary": {
            "BillingMode": "PAY_PER_REQUEST"
        }
    }
}
```


Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 3 : pour créer une table et la chiffrer à l'aide d'une clé CMK gérée par le client

L'exemple suivant crée une table nommée `MusicCollection` et la chiffre à l'aide d'une clé CMK gérée par le client.

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=abcd1234-abcd-1234-
a123-ab1234a1b234
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-27T11:12:16.431000-07:00",
```

```
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "SSEDescription": {
            "Status": "ENABLED",
            "SSEType": "KMS",
            "KMSMasterKeyArn": "arn:aws:kms:us-west-2:123456789012:key/abcd1234-
abcd-1234-a123-ab1234a1b234"
        }
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 4 : pour créer une table avec un index secondaire local

L'exemple suivant utilise les attributs et le schéma de clés spécifiés pour créer une table nommée `MusicCollection` avec un index secondaire local nommé`AlbumTitleIndex`.

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S Att
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --local-secondary-indexes \
        "[
            {
                \"IndexName\": \"AlbumTitleIndex\",
                \"KeySchema\": [
                    {\"AttributeName\": \"Artist\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\": \"AlbumTitle\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {
```

```
                    \"ProjectionType\": \"INCLUDE\",
                    \"NonKeyAttributes\": [\"Genre\", \"Year\"]
                }
            }
        ]"
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
```

```
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "LocalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "Artist",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "Genre",
                        "Year"
                    ]
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 5 : pour créer une table avec un index secondaire global

L'exemple suivant crée une table nommée `GameScores` avec un index secondaire global appelé`GameTitleIndex`. La table de base a une clé de partition `UserId` et une clé de tri `GameTitle`, vous permettant de trouver efficacement le meilleur score d'un utilisateur pour un jeu spécifique, tandis que l'index secondaire global (GSI) a une clé de partition `GameTitle` et une clé de tri `TopScore`, vous permettant de trouver rapidement le score le plus élevé pour un jeu particulier.

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
 \
    --key-schema AttributeName=UserId,KeyType=HASH \
                 AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"GameTitleIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"GameTitle\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"TopScore\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {
                    \"ProjectionType\":\"INCLUDE\",
                    \"NonKeyAttributes\":[\"UserId\"]
                },
                \"ProvisionedThroughput\": {
                    \"ReadCapacityUnits\": 10,
                    \"WriteCapacityUnits\": 5
                }
            }
        ]"
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "TopScore",
                "AttributeType": "N"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
```

```
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-26T17:28:15.602000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "GameTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "GameTitle",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "TopScore",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "UserId"
                    ]
                },
                "IndexStatus": "CREATING",
```

```
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 5
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 6 : pour créer une table avec plusieurs index secondaires globaux à la fois

L'exemple suivant crée une table nommée `GameScores` avec deux index secondaires globaux. Les schémas GSI sont transmis via un fichier plutôt que sur la ligne de commande.

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --global-secondary-indexes file://gsi.json
```

Contenu de `gsi.json` :

```
[
    {
        "IndexName": "GameTitleIndex",
        "KeySchema": [
            {
                "AttributeName": "GameTitle",
                "KeyType": "HASH"
            },
```

```
            {
                "AttributeName": "TopScore",
                "KeyType": "RANGE"
            }
        ],
        "Projection": {
            "ProjectionType": "ALL"
        },
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        }
    },
    {
        "IndexName": "GameDateIndex",
        "KeySchema": [
            {
                "AttributeName": "GameTitle",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "Date",
                "KeyType": "RANGE"
            }
        ],
        "Projection": {
            "ProjectionType": "ALL"
        },
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        }
    }
]
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Date",
                "AttributeType": "S"
```

```
            },
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "TopScore",
                "AttributeType": "N"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-08-04T16:40:55.524000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "GameTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "GameTitle",
                        "KeyType": "HASH"
                    },
```

```
                    {
                        "AttributeName": "TopScore",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "CREATING",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 5
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
            },
            {
                "IndexName": "GameDateIndex",
                "KeySchema": [
                    {
                        "AttributeName": "GameTitle",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "Date",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "CREATING",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 5
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameDateIndex"
```

```
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 7 : pour créer une table avec Streams activé

L'exemple suivant crée une table appelée `GameScores` avec DynamoDB Streams activé. Les nouvelles et les anciennes images de chaque article seront écrites dans le flux.

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --stream-specification StreamEnabled=TRUE,StreamViewType=NEW_AND_OLD_IMAGES
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
```

```
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-27T10:49:34.056000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_AND_OLD_IMAGES"
        },
        "LatestStreamLabel": "2020-05-27T17:49:34.056",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2020-05-27T17:49:34.056"
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour les tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 8 : pour créer une table avec Keys-Only Stream activé

L'exemple suivant crée une table appelée `GameScores` avec DynamoDB Streams activé. Seuls les attributs clés des éléments modifiés sont écrits dans le flux.

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --stream-specification StreamEnabled=TRUE,StreamViewType=KEYS_ONLY
```

Sortie :


```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T18:45:34.140000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "KEYS_ONLY"
        },
        "LatestStreamLabel": "2023-05-25T18:45:34.140",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2023-05-25T18:45:34.140",
        "DeletionProtectionEnabled": false
```

```
    }
}
```

Pour plus d'informations, consultez Modifier la capture de données pour DynamoDB Streams dans le manuel Amazon DynamoDB Developer Guide.

Exemple 9 : pour créer une table avec la classe Standard Infrequent Access

L'exemple suivant crée une table appelée `GameScores` et affecte la classe de table Standard-Infrequent Access (DynamoDB Standard-IA). Cette classe de table est optimisée pour le stockage, qui constitue le principal coût.

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --table-class STANDARD_INFREQUENT_ACCESS
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
```

```
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T18:33:07.581000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "TableClassSummary": {
            "TableClass": "STANDARD_INFREQUENT_ACCESS"
        },
        "DeletionProtectionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Classes de tables dans le manuel Amazon DynamoDB Developer Guide.

Exemple 10 : pour créer une table avec la protection contre la suppression activée

L'exemple suivant crée une table appelée `GameScores` et active la protection contre la suppression.

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --deletion-protection-enabled
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T23:02:17.093000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeletionProtectionEnabled": true
    }
}
```


Pour plus d'informations, consultez la section Utilisation de la protection contre les suppressions dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateTableà la section Référence des AWS CLI commandes.

`delete-backup`

L'exemple de code suivant montre comment utiliser`delete-backup`.

AWS CLI

Pour supprimer une sauvegarde DynamoDB existante

L'`delete-backup`exemple suivant supprime la sauvegarde existante spécifiée.

```
aws dynamodb delete-backup \
    --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

Sortie :

```
{
    "BackupDescription": {
        "BackupDetails": {
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
            "BackupName": "MusicCollectionBackup",
            "BackupSizeBytes": 0,
            "BackupStatus": "DELETED",
            "BackupType": "USER",
            "BackupCreationDateTime": 1576616366.715
        },
        "SourceTableDetails": {
            "TableName": "MusicCollection",
            "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "TableSizeBytes": 0,
            "KeySchema": [
                {
                    "AttributeName": "Artist",
                    "KeyType": "HASH"
                },
                {
                    "AttributeName": "SongTitle",
                    "KeyType": "RANGE"
                }
            ],
```

```
            "TableCreationDateTime": 1576615228.571,
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
            },
            "ItemCount": 0,
            "BillingMode": "PROVISIONED"
        },
        "SourceTableFeatureDetails": {}
    }
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DeleteBackupà la section Référence des AWS CLI commandes.

`delete-item`

L'exemple de code suivant montre comment utiliser`delete-item`.

AWS CLI

Exemple 1 : pour supprimer un élément

L'`delete-item`exemple suivant supprime un élément du `MusicCollection` tableau et demande des informations sur l'élément supprimé et sur la capacité utilisée par la demande.

```
aws dynamodb delete-item \
    --table-name MusicCollection \
    --key file://key.json \
    --return-values ALL_OLD \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

Contenu de `key.json` :

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Scared of My Shadow"}
}
```

Sortie :

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Blue Sky Blues"
        },
        "Artist": {
            "S": "No One You Know"
        },
        "SongTitle": {
            "S": "Scared of My Shadow"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 2.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
            "Artist": {
                "S": "No One You Know"
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

Pour plus d'informations, consultez la section Rédaction d'un article dans le guide du développeur Amazon DynamoDB.

Exemple 2 : pour supprimer un élément de manière conditionnelle

L'exemple suivant supprime un article du `ProductCatalog` tableau uniquement s'il se situe entre 500 `Gardening Supplies` et 600 `ProductCategory` € `Sporting Goods` ou si son prix est compris entre 500 et 600€. Elle renvoie des informations sur l'élément qui a été supprimé.

```
aws dynamodb delete-item \
    --table-name ProductCatalog \
    --key '{"Id":{"N":"456"}}' \
```

```
    --condition-expression "(ProductCategory IN (:cat1, :cat2)) and (#P between :lo
 and :hi)" \
    --expression-attribute-names file://names.json \
    --expression-attribute-values file://values.json \
    --return-values ALL_OLD
```

Contenu de `names.json` :

```
{
    "#P": "Price"
}
```

Contenu de `values.json` :

```
{
    ":cat1": {"S": "Sporting Goods"},
    ":cat2": {"S": "Gardening Supplies"},
    ":lo": {"N": "500"},
    ":hi": {"N": "600"}
}
```

Sortie :

```
{
    "Attributes": {
        "Id": {
            "N": "456"
        },
        "Price": {
            "N": "550"
        },
        "ProductCategory": {
            "S": "Sporting Goods"
        }
    }
}
```

Pour plus d'informations, consultez la section Rédaction d'un article dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DeleteItemà la section Référence des AWS CLI commandes.

`delete-table`

L'exemple de code suivant montre comment utiliser`delete-table`.

AWS CLI

Pour supprimer un tableau

L'`delete-table`exemple suivant supprime le `MusicCollection` tableau.

```
aws dynamodb delete-table \
    --table-name MusicCollection
```

Sortie :

```
{
    "TableDescription": {
        "TableStatus": "DELETING",
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableName": "MusicCollection",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        }
    }
}
```

Pour plus d'informations, consultez Supprimer une table dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteTableà la section Référence des AWS CLI commandes.

`describe-backup`

L'exemple de code suivant montre comment utiliser`describe-backup`.

AWS CLI

Pour obtenir des informations sur une sauvegarde existante d'une table

L'`describe-backup`exemple suivant affiche des informations sur la sauvegarde existante spécifiée.

```
aws dynamodb describe-backup \
    --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/backup/01576616366715-b4e58d3a
```

Sortie :

```
{
    "BackupDescription": {
        "BackupDetails": {
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/backup/01576616366715-b4e58d3a",
            "BackupName": "MusicCollectionBackup",
            "BackupSizeBytes": 0,
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupCreationDateTime": 1576616366.715
        },
        "SourceTableDetails": {
            "TableName": "MusicCollection",
            "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
            "TableSizeBytes": 0,
            "KeySchema": [
                {
                    "AttributeName": "Artist",
                    "KeyType": "HASH"
                },
                {
                    "AttributeName": "SongTitle",
                    "KeyType": "RANGE"
                }
            ],
            "TableCreationDateTime": 1576615228.571,
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
            },
            "ItemCount": 0,
            "BillingMode": "PROVISIONED"
```

```
        },
        "SourceTableFeatureDetails": {}
    }
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeBackupà la section Référence des AWS CLI commandes.

`describe-continuous-backups`

L'exemple de code suivant montre comment utiliser`describe-continuous-backups`.

AWS CLI

Pour obtenir des informations sur les sauvegardes continues d'une table DynamoDB

L'`describe-continuous-backups`exemple suivant affiche des détails sur les paramètres de sauvegarde continue de la `MusicCollection` table.

```
aws dynamodb describe-continuous-backups \
    --table-name MusicCollection
```

Sortie :

```
{
    "ContinuousBackupsDescription": {
        "ContinuousBackupsStatus": "ENABLED",
        "PointInTimeRecoveryDescription": {
            "PointInTimeRecoveryStatus": "DISABLED"
        }
    }
}
```

Pour plus d'informations, consultez la section Point-in-Time Recovery for DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeContinuousBackupsà la section Référence des AWS CLI commandes.

`describe-contributor-insights`

L'exemple de code suivant montre comment utiliser`describe-contributor-insights`.

AWS CLI

Pour afficher les paramètres de Contributor Insights pour une table DynamoDB

L'`describe-contributor-insights`exemple suivant affiche les paramètres de Contributor Insights pour la `MusicCollection` table et l'index secondaire `AlbumTitle-index` global.

```
aws dynamodb describe-contributor-insights \
    --table-name MusicCollection \
    --index-name AlbumTitle-index
```

Sortie :

```
{
    "TableName": "MusicCollection",
    "IndexName": "AlbumTitle-index",
    "ContributorInsightsRuleList": [
        "DynamoDBContributorInsights-PKC-MusicCollection-1576629651520",
        "DynamoDBContributorInsights-SKC-MusicCollection-1576629651520",
        "DynamoDBContributorInsights-PKT-MusicCollection-1576629651520",
        "DynamoDBContributorInsights-SKT-MusicCollection-1576629651520"
    ],
    "ContributorInsightsStatus": "ENABLED",
    "LastUpdateDateTime": 1576629654.78
}
```

Pour plus d'informations, consultez la section Analyse de l'accès aux données à l'aide de CloudWatch Contributor Insights for DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeContributorInsightsà la section Référence des AWS CLI commandes.

`describe-endpoints`

L'exemple de code suivant montre comment utiliser`describe-endpoints`.

AWS CLI

Pour consulter les informations relatives aux terminaux régionaux

L'`describe-endpoints`exemple suivant affiche des détails sur les points de terminaison de la AWS région actuelle.

```
aws dynamodb describe-endpoints
```

Sortie :

```
{
    "Endpoints": [
        {
            "Address": "dynamodb.us-west-2.amazonaws.com",
            "CachePeriodInMinutes": 1440
        }
    ]
}
```

Pour plus d'informations, consultez la section Points de terminaison et quotas Amazon DynamoDB dans le manuel de référence général.AWS

- Pour plus de détails sur l'API, reportez-vous DescribeEndpointsà la section Référence des AWS CLI commandes.

`describe-global-table-settings`

L'exemple de code suivant montre comment utiliser`describe-global-table-settings`.

AWS CLI

Pour obtenir des informations sur les paramètres d'une table globale DynamoDB

L'`describe-global-table-settings`exemple suivant montre les paramètres de la table `MusicCollection` globale.

```
aws dynamodb describe-global-table-settings \
    --global-table-name MusicCollection
```

Sortie :

```
{
    "GlobalTableName": "MusicCollection",
    "ReplicaSettings": [
        {
            "RegionName": "us-east-1",
            "ReplicaStatus": "ACTIVE",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 5,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        },
        {
            "RegionName": "us-east-2",
            "ReplicaStatus": "ACTIVE",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 5,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        }
    ]
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeGlobalTableSettingsà la section Référence des AWS CLI commandes.

`describe-global-table`

L'exemple de code suivant montre comment utiliser`describe-global-table`.

AWS CLI

Pour afficher les informations relatives à une table globale DynamoDB

L'`describe-global-table`exemple suivant affiche les détails de la table `MusicCollection` globale.

```
aws dynamodb describe-global-table \
    --global-table-name MusicCollection
```

Sortie :

```
{
    "GlobalTableDescription": {
        "ReplicationGroup": [
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
        "CreationDateTime": 1576625818.532,
        "GlobalTableStatus": "ACTIVE",
        "GlobalTableName": "MusicCollection"
    }
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeGlobalTableà la section Référence des AWS CLI commandes.

`describe-limits`

L'exemple de code suivant montre comment utiliser`describe-limits`.

AWS CLI

Pour consulter les limites de capacité provisionnée

L'`describe-limits`exemple suivant affiche les limites de capacité allouées pour votre compte dans la région actuelle. AWS

```
aws dynamodb describe-limits
```

Sortie :

```
{
    "AccountMaxReadCapacityUnits": 80000,
    "AccountMaxWriteCapacityUnits": 80000,
    "TableMaxReadCapacityUnits": 40000,
    "TableMaxWriteCapacityUnits": 40000
}
```

Pour plus d'informations, consultez la section Limites dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeLimitsà la section Référence des AWS CLI commandes.

`describe-table-replica-auto-scaling`

L'exemple de code suivant montre comment utiliser`describe-table-replica-auto-scaling`.

AWS CLI

Pour afficher les paramètres de mise à l'échelle automatique sur les répliques d'une table globale

L'`describe-table-replica-auto-scaling`exemple suivant affiche les paramètres de mise à l'échelle automatique entre les répliques de la table `MusicCollection` globale.

```
aws dynamodb describe-table-replica-auto-scaling \
    --table-name MusicCollection
```

Sortie :

```
{
    "TableAutoScalingDescription": {
        "TableName": "MusicCollection",
        "TableStatus": "ACTIVE",
        "Replicas": [
            {
                "RegionName": "us-east-1",
                "GlobalSecondaryIndexes": [],
```

```
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            },
            {
                "RegionName": "us-east-2",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
```

```
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            }
        ]
    }
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeTableReplicaAutoScalingà la section Référence des AWS CLI commandes.

`describe-table`

L'exemple de code suivant montre comment utiliser`describe-table`.

AWS CLI

Pour décrire un tableau

L'`describe-table`exemple suivant décrit le `MusicCollection` tableau.

```
aws dynamodb describe-table \
    --table-name MusicCollection
```

Sortie :

```
{
    "Table": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "TableName": "MusicCollection",
        "TableStatus": "ACTIVE",
        "KeySchema": [
            {
                "KeyType": "HASH",
                "AttributeName": "Artist"
            },
            {
                "KeyType": "RANGE",
                "AttributeName": "SongTitle"
            }
        ],
        "ItemCount": 0,
        "CreationDateTime": 1421866952.062
```

```
    }
}
```

Pour plus d'informations, consultez la section Description d'une table dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeTableà la section Référence des AWS CLI commandes.

`describe-time-to-live`

L'exemple de code suivant montre comment utiliser`describe-time-to-live`.

AWS CLI

Pour afficher les paramètres Time to Live d'une table

L'`describe-time-to-live`exemple suivant affiche les paramètres Time to Live de la `MusicCollection` table.

```
aws dynamodb describe-time-to-live \
    --table-name MusicCollection
```

Sortie :

```
{
    "TimeToLiveDescription": {
        "TimeToLiveStatus": "ENABLED",
        "AttributeName": "ttl"
    }
}
```

Pour plus d'informations, consultez Time to Live dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous DescribeTimeToLiveà la section Référence des AWS CLI commandes.

`get-item`

L'exemple de code suivant montre comment utiliser`get-item`.

AWS CLI

Exemple 1 : Pour lire un élément dans un tableau

L'`get-item`exemple suivant extrait un élément du `MusicCollection` tableau. La table possède une clé hash-and-range primaire (`Artist`et`SongTitle`), vous devez donc spécifier ces deux attributs. La commande demande également des informations sur la capacité de lecture consommée par l'opération.

```
aws dynamodb get-item \
    --table-name MusicCollection \
    --key file://key.json \
    --return-consumed-capacity TOTAL
```

Contenu de `key.json` :

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

Sortie :

```
{
    "Item": {
        "AlbumTitle": {
            "S": "Songs About Life"
        },
        "SongTitle": {
            "S": "Happy Day"
        },
        "Artist": {
            "S": "Acme Band"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

Pour plus d'informations, consultez Lire un article dans le guide du développeur Amazon DynamoDB.

Exemple 2 : Pour lire un élément en utilisant une lecture cohérente

L'exemple suivant extrait un élément du `MusicCollection` tableau à l'aide de lectures très cohérentes.

```
aws dynamodb get-item \
    --table-name MusicCollection \
    --key file://key.json \
    --consistent-read \
    --return-consumed-capacity TOTAL
```

Contenu de `key.json` :

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

Sortie :

```
{
    "Item": {
        "AlbumTitle": {
            "S": "Songs About Life"
        },
        "SongTitle": {
            "S": "Happy Day"
        },
        "Artist": {
            "S": "Acme Band"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}
```

Pour plus d'informations, consultez Lire un article dans le guide du développeur Amazon DynamoDB.

Exemple 3 : Pour récupérer les attributs spécifiques d'un article

L'exemple suivant utilise une expression de projection pour récupérer uniquement trois attributs de l'élément souhaité.

```
aws dynamodb get-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "102"}}' \
    --projection-expression "#T, #C, #P" \
    --expression-attribute-names file://names.json
```

Contenu de `names.json` :

```
{
    "#T": "Title",
    "#C": "ProductCategory",
    "#P": "Price"
}
```

Sortie :

```
{
    "Item": {
        "Price": {
            "N": "20"
        },
        "Title": {
            "S": "Book 102 Title"
        },
        "ProductCategory": {
            "S": "Book"
        }
    }
}
```

Pour plus d'informations, consultez Lire un article dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous GetItemà la section Référence des AWS CLI commandes.

`list-backups`

L'exemple de code suivant montre comment utiliser`list-backups`.

AWS CLI

Exemple 1 : pour répertorier toutes les sauvegardes DynamoDB existantes

L'`list-backups`exemple suivant répertorie toutes vos sauvegardes existantes.

```
aws dynamodb list-backups
```

Sortie :

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        },
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
            "BackupName": "MusicCollectionBackup2",
            "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
            "BackupStatus": "AVAILABLE",
```

```
            "BackupType": "USER",
            "BackupSizeBytes": 400
        }
    ]
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 2 : pour répertorier les sauvegardes créées par l'utilisateur dans un intervalle de temps spécifique

L'exemple suivant répertorie uniquement les sauvegardes de la `MusicCollection` table créées par l'utilisateur (et non celles créées automatiquement par DynamoDB) dont la date de création est comprise entre le 1er janvier 2020 et le 1er mars 2020.

```
aws dynamodb list-backups \
    --table-name MusicCollection \
    --time-range-lower-bound 1577836800 \
    --time-range-upper-bound 1583020800 \
    --backup-type USER
```

Sortie :

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        }
    ]
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 3 : pour limiter la taille de page

L'exemple suivant renvoie une liste de toutes les sauvegardes existantes, mais récupère un seul élément par appel, en effectuant plusieurs appels si nécessaire pour obtenir la liste complète. La limitation de la taille de page est utile lorsque vous exécutez des commandes de liste sur un grand nombre de ressources, ce qui peut entraîner une erreur de « expiration du délai imparti » lors de l'utilisation du format de page par défaut de 1 000.

```
aws dynamodb list-backups \
    --page-size 1
```

Sortie :

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        },
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
            "BackupName": "MusicCollectionBackup2",
            "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
```

```
            "BackupSizeBytes": 400
        }
    ]
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 4 : Pour limiter le nombre d'articles renvoyés

L'exemple suivant limite le nombre d'éléments renvoyés à 1. La réponse inclut une `NextToken` valeur permettant de récupérer la page de résultats suivante.

```
aws dynamodb list-backups \
    --max-items 1
```

Sortie :

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 5 : Pour récupérer la page de résultats suivante

La commande suivante utilise la `NextToken` valeur d'un appel précédent à la `list-backups` commande pour récupérer une autre page de résultats. Dans ce cas, étant donné que la réponse n'inclut aucune `NextToken` valeur, nous savons que nous avons atteint la fin des résultats.

```
aws dynamodb list-backups \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Sortie

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
            "BackupName": "MusicCollectionBackup2",
            "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 400
        }
    ]
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous ListBackupsà la section Référence des AWS CLI commandes.

`list-contributor-insights`

L'exemple de code suivant montre comment utiliser`list-contributor-insights`.

AWS CLI

Exemple 1 : pour afficher une liste de résumés de Contributor Insights

L'`list-contributor-insights`exemple suivant affiche une liste de résumés de Contributor Insights.

```
aws dynamodb list-contributor-insights
```

Sortie :

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "MusicCollection",
            "IndexName": "AlbumTitle-index",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "ProductCatalog",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Forum",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Reply",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Thread",
            "ContributorInsightsStatus": "ENABLED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse de l'accès aux données à l'aide de CloudWatch Contributor Insights for DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 2 : pour limiter le nombre d'articles renvoyés

L'exemple suivant limite le nombre d'articles renvoyés à 4. La réponse inclut une `NextToken` valeur permettant de récupérer la page de résultats suivante.

```
aws dynamodb list-contributor-insights \
    --max-results 4
```

Sortie :

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "MusicCollection",
            "IndexName": "AlbumTitle-index",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "ProductCatalog",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Forum",
            "ContributorInsightsStatus": "ENABLED"
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Pour plus d'informations, consultez la section Analyse de l'accès aux données à l'aide de CloudWatch Contributor Insights for DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 3 : Pour récupérer la page de résultats suivante

La commande suivante utilise la `NextToken` valeur d'un appel précédent à la `list-contributor-insights` commande pour récupérer une autre page de résultats. Dans ce cas, étant donné que la réponse n'inclut aucune `NextToken` valeur, nous savons que nous avons atteint la fin des résultats.

```
aws dynamodb list-contributor-insights \
    --max-results 4 \
    --next-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Sortie :

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "Reply",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Thread",
            "ContributorInsightsStatus": "ENABLED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse de l'accès aux données à l'aide de CloudWatch Contributor Insights for DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous ListContributorInsightsà la section Référence des AWS CLI commandes.

`list-global-tables`

L'exemple de code suivant montre comment utiliser`list-global-tables`.

AWS CLI

Pour répertorier les tables globales DynamoDB existantes

L'`list-global-tables`exemple suivant répertorie toutes vos tables globales existantes.

```
aws dynamodb list-global-tables
```

Sortie :

```
{
    "GlobalTables": [
        {
            "GlobalTableName": "MusicCollection",
            "ReplicationGroup": [
                {
```

```
                    "RegionName": "us-east-2"
                },
                {
                    "RegionName": "us-east-1"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous ListGlobalTablesà la section Référence des AWS CLI commandes.

`list-tables`

L'exemple de code suivant montre comment utiliser`list-tables`.

AWS CLI

Exemple 1 : pour répertorier les tables

L'`list-tables`exemple suivant répertorie toutes les tables associées au AWS compte courant et à la région.

```
aws dynamodb list-tables
```

Sortie :

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog",
        "Reply",
        "Thread"
    ]
}
```

Pour plus d'informations, consultez Listing Table Names dans le manuel Amazon DynamoDB Developer Guide.

Exemple 2 : pour limiter la taille de page

L'exemple suivant renvoie une liste de toutes les tables existantes, mais extrait un seul élément par appel, en effectuant plusieurs appels si nécessaire pour obtenir la liste complète. La limitation de la taille de page est utile lorsque vous exécutez des commandes de liste sur un grand nombre de ressources, ce qui peut entraîner une erreur de « expiration du délai imparti » lors de l'utilisation du format de page par défaut de 1 000.

```
aws dynamodb list-tables \
    --page-size 1
```

Sortie :

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog",
        "Reply",
        "Thread"
    ]
}
```

Pour plus d'informations, consultez Listing Table Names dans le manuel Amazon DynamoDB Developer Guide.

Exemple 3 : pour limiter le nombre d'articles renvoyés

L'exemple suivant limite le nombre d'articles renvoyés à 2. La réponse inclut une `NextToken` valeur permettant de récupérer la page de résultats suivante.

```
aws dynamodb list-tables \
    --max-items 2
```

Sortie :

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog"
```

```
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Pour plus d'informations, consultez Listing Table Names dans le manuel Amazon DynamoDB Developer Guide.

Exemple 4 : Pour récupérer la page de résultats suivante

La commande suivante utilise la `NextToken` valeur d'un appel précédent à la `list-tables` commande pour récupérer une autre page de résultats. Dans ce cas, étant donné que la réponse n'inclut aucune `NextToken` valeur, nous savons que nous avons atteint la fin des résultats.

```
aws dynamodb list-tables \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Sortie :

```
{
    "TableNames": [
        "Reply",
        "Thread"
    ]
}
```

Pour plus d'informations, consultez Listing Table Names dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTablesà la section Référence des AWS CLI commandes.

`list-tags-of-resource`

L'exemple de code suivant montre comment utiliser`list-tags-of-resource`.

AWS CLI

Exemple 1 : pour répertorier les balises d'une ressource DynamoDB

L'`list-tags-of-resource`exemple suivant montre comment afficher les balises du `MusicCollection` tableau.

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Owner",
            "Value": "blueTeam"
        },
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ]
}
```

Pour plus d'informations, consultez la section Balisage pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 2 : pour limiter le nombre de balises renvoyées

L'exemple suivant limite le nombre de balises renvoyées à 1. La réponse inclut une `NextToken` valeur permettant de récupérer la page de résultats suivante.

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --max-items 1
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Owner",
            "Value": "blueTeam"
        }
```

```
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Pour plus d'informations, consultez la section Balisage pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 3 : Pour récupérer la page de résultats suivante

La commande suivante utilise la `NextToken` valeur d'un appel précédent à la `list-tags-of-resource` commande pour récupérer une autre page de résultats. Dans ce cas, étant donné que la réponse n'inclut aucune `NextToken` valeur, nous savons que nous avons atteint la fin des résultats.

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ]
}
```

Pour plus d'informations, consultez la section Balisage pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous ListTagsOfResourceà la section Référence des AWS CLI commandes.

`put-item`

L'exemple de code suivant montre comment utiliser`put-item`.

AWS CLI

Exemple 1 : pour ajouter un élément à un tableau

L'`put-item`exemple suivant ajoute un nouvel élément au MusicCollectiontableau.

```
aws dynamodb put-item \
    --table-name MusicCollection \
    --item file://item.json \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

Contenu de `item.json` :

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Call Me Today"},
    "AlbumTitle": {"S": "Greatest Hits"}
}
```

Sortie :

```
{
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
            "Artist": {
                "S": "No One You Know"
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

Pour plus d'informations, consultez la section Rédaction d'un article dans le guide du développeur Amazon DynamoDB.

Exemple 2 : Pour remplacer de manière conditionnelle un élément d'un tableau

L'`put-item`exemple suivant remplace un élément existant dans le `MusicCollection` tableau uniquement si cet élément possède un `AlbumTitle` attribut dont la valeur est de`Greatest Hits`. La commande renvoie la valeur précédente de l'élément.

```
aws dynamodb put-item \
    --table-name MusicCollection \
    --item file://item.json \
    --condition-expression "#A = :A" \
    --expression-attribute-names file://names.json \
    --expression-attribute-values file://values.json \
    --return-values ALL_OLD
```

Contenu de `item.json` :

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Call Me Today"},
    "AlbumTitle": {"S": "Somewhat Famous"}
}
```

Contenu de `names.json` :

```
{
    "#A": "AlbumTitle"
}
```

Contenu de `values.json` :

```
{
    ":A": {"S": "Greatest Hits"}
}
```

Sortie :

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Greatest Hits"
        },
```

```
        "Artist": {
            "S": "No One You Know"
        },
        "SongTitle": {
            "S": "Call Me Today"
        }
    }
}
```

Si la clé existe déjà, vous devriez voir le résultat suivant :

```
A client error (ConditionalCheckFailedException) occurred when calling the PutItem
 operation: The conditional request failed.
```

Pour plus d'informations, consultez la section Rédaction d'un article dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous PutItemà la section Référence des AWS CLI commandes.

`query`

L'exemple de code suivant montre comment utiliser`query`.

AWS CLI

Exemple 1 : pour interroger une table

L'`query`exemple suivant interroge les éléments de la `MusicCollection` table. La table possède une clé hash-and-range primaire (`Artist`et`SongTitle`), mais cette requête indique uniquement la valeur de la clé de hachage. Il renvoie les titres des chansons de l'artiste nommées « No One You Know ».

```
aws dynamodb query \
    --table-name MusicCollection \
    --projection-expression "SongTitle" \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json \
    --return-consumed-capacity TOTAL
```

Contenu de `expression-attributes.json` :

```
{
    ":v1": {"S": "No One You Know"}
}
```

Sortie :

```
{
    "Items": [
        {
            "SongTitle": {
                "S": "Call Me Today"
            },
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

Pour plus d'informations, consultez la section Utilisation des requêtes dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 2 : pour interroger une table à l'aide de lectures très cohérentes et parcourir l'index par ordre décroissant

L'exemple suivant exécute la même requête que le premier exemple, mais renvoie les résultats dans l'ordre inverse et utilise des lectures très cohérentes.

```
aws dynamodb query \
    --table-name MusicCollection \
    --projection-expression "SongTitle" \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json \
    --consistent-read \
    --no-scan-index-forward \
    --return-consumed-capacity TOTAL
```

Contenu de `expression-attributes.json` :

```
{
    ":v1": {"S": "No One You Know"}
}
```

Sortie :

```
{
    "Items": [
        {
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        },
        {
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}
```

Pour plus d'informations, consultez la section Utilisation des requêtes dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 3 : pour filtrer des résultats spécifiques

L'exemple suivant interroge les résultats `MusicCollection` mais exclut les résultats dont l'`AlbumTitle`attribut contient des valeurs spécifiques. Notez que cela n'affecte pas le `ScannedCount` ou`ConsumedCapacity`, car le filtre est appliqué après la lecture des éléments.

```
aws dynamodb query \
    --table-name MusicCollection \
    --key-condition-expression "#n1 = :v1" \
    --filter-expression "NOT (#n2 IN (:v2, :v3))" \
```

```
    --expression-attribute-names file://names.json \
    --expression-attribute-values file://values.json \
    --return-consumed-capacity TOTAL
```

Contenu de `values.json` :

```
{
    ":v1": {"S": "No One You Know"},
    ":v2": {"S": "Blue Sky Blues"},
    ":v3": {"S": "Greatest Hits"}
}
```

Contenu de `names.json` :

```
{
    "#n1": "Artist",
    "#n2": "AlbumTitle"
}
```

Sortie :

```
{
    "Items": [
        {
            "AlbumTitle": {
                "S": "Somewhat Famous"
            },
            "Artist": {
                "S": "No One You Know"
            },
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 1,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

Pour plus d'informations, consultez la section Utilisation des requêtes dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 4 : Pour récupérer uniquement le nombre d'articles

L'exemple suivant récupère le nombre d'éléments correspondant à la requête, mais ne récupère aucun des éléments eux-mêmes.

```
aws dynamodb query \
    --table-name MusicCollection \
    --select COUNT \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json
```

Contenu de `expression-attributes.json` :

```
{
    ":v1": {"S": "No One You Know"}
}
```

Sortie :

```
{
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": null
}
```

Pour plus d'informations, consultez la section Utilisation des requêtes dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 5 : Pour interroger un index

L'exemple suivant interroge l'index secondaire local`AlbumTitleIndex`. La requête renvoie tous les attributs de la table de base qui ont été projetés dans l'index secondaire local. Notez que lorsque vous interrogez un index secondaire local ou un index secondaire global, vous devez également fournir le nom de la table de base à l'aide du `table-name` paramètre.

```
aws dynamodb query \
    --table-name MusicCollection \
    --index-name AlbumTitleIndex \
    --key-condition-expression "Artist = :v1" \
```

```
    --expression-attribute-values file://expression-attributes.json \
    --select ALL_PROJECTED_ATTRIBUTES \
    --return-consumed-capacity INDEXES
```

Contenu de `expression-attributes.json` :

```
{
    ":v1": {"S": "No One You Know"}
}
```

Sortie :

```
{
    "Items": [
        {
            "AlbumTitle": {
                "S": "Blue Sky Blues"
            },
            "Artist": {
                "S": "No One You Know"
            },
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        },
        {
            "AlbumTitle": {
                "S": "Somewhat Famous"
            },
            "Artist": {
                "S": "No One You Know"
            },
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5,
        "Table": {
```

```
            "CapacityUnits": 0.0
        },
        "LocalSecondaryIndexes": {
            "AlbumTitleIndex": {
                "CapacityUnits": 0.5
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Utilisation des requêtes dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus d'informations sur l'API, consultez Query dans la Référence des commandes AWS CLI .

`restore-table-from-backup`

L'exemple de code suivant montre comment utiliser`restore-table-from-backup`.

AWS CLI

Pour restaurer une table DynamoDB à partir d'une sauvegarde existante

L'`restore-table-from-backup`exemple suivant restaure la table spécifiée à partir d'une sauvegarde existante.

```
aws dynamodb restore-table-from-backup \
    --target-table-name MusicCollection \
    --backup-arnarn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
```

```
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection2",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": 1576618274.326,
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection2",
        "TableId": "114865c9-5ef3-496c-b4d1-c4cbdd2d44fb",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED"
        },
        "RestoreSummary": {
            "SourceBackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
            "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "RestoreDateTime": 1576616366.715,
            "RestoreInProgress": true
        }
    }
}
```

Pour plus d'informations, consultez la section Sauvegarde et restauration à la demande pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous RestoreTableFromBackupà la section Référence des AWS CLI commandes.

`restore-table-to-point-in-time`

L'exemple de code suivant montre comment utiliser`restore-table-to-point-in-time`.

AWS CLI

Pour restaurer une table DynamoDB à un point précis dans le temps

L'`restore-table-to-point-in-time`exemple suivant restaure la `MusicCollection` table à l'instant spécifié.

```
aws dynamodb restore-table-to-point-in-time \
    --source-table-name MusicCollection \
    --target-table-name MusicCollectionRestore \
    --restore-date-time 1576622404.0
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollectionRestore",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
```

```
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": 1576623311.86,
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollectionRestore",
        "TableId": "befd9e0e-1843-4dc6-a147-d6d00e85cb1f",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED"
        },
        "RestoreSummary": {
            "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "RestoreDateTime": 1576622404.0,
            "RestoreInProgress": true
        }
    }
}
```

Pour plus d'informations, consultez la section Point-in-Time Recovery for DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous RestoreTableToPointInTimeà la section Référence des AWS CLI commandes.

`scan`

L'exemple de code suivant montre comment utiliser`scan`.

AWS CLI

Pour scanner un tableau

L'`scan`exemple suivant analyse l'intégralité du `MusicCollection` tableau, puis réduit les résultats aux chansons de l'artiste « No One You Know ». Pour chaque élément, seuls le titre de l'album et le titre de la chanson sont renvoyés.

```
aws dynamodb scan \
    --table-name MusicCollection \
    --filter-expression "Artist = :a" \
    --projection-expression "#ST, #AT" \
    --expression-attribute-names file://expression-attribute-names.json \
    --expression-attribute-values file://expression-attribute-values.json
```

Contenu de `expression-attribute-names.json` :

```
{
    "#ST": "SongTitle",
    "#AT":"AlbumTitle"
}
```

Contenu de `expression-attribute-values.json` :

```
{
    ":a": {"S": "No One You Know"}
}
```

Sortie :

```
{
    "Count": 2,
    "Items": [
        {
            "SongTitle": {
                "S": "Call Me Today"
            },
            "AlbumTitle": {
                "S": "Somewhat Famous"
            }
        },
        {
            "SongTitle": {
                "S": "Scared of My Shadow"
            },
            "AlbumTitle": {
                "S": "Blue Sky Blues"
            }
        }
```

```
    ],
    "ScannedCount": 3,
    "ConsumedCapacity": null
}
```

Pour plus d'informations, consultez la section Utilisation des scans dans DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus d'informations sur l'API, consultez Scan dans la Référence des commandes AWS CLI .

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une ressource DynamoDB

L'`tag-resource`exemple suivant ajoute une key/value paire de balises au `MusicCollection` tableau.

```
aws dynamodb tag-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --tags Key=Owner,Value=blueTeam
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`transact-get-items`

L'exemple de code suivant montre comment utiliser`transact-get-items`.

AWS CLI

Pour récupérer plusieurs éléments de manière atomique à partir d'une ou de plusieurs tables

L'`transact-get-items`exemple suivant récupère plusieurs éléments de manière atomique.

```
aws dynamodb transact-get-items \
    --transact-items file://transact-items.json \
    --return-consumed-capacity TOTAL
```

Contenu de `transact-items.json` :

```
[
    {
        "Get": {
            "Key": {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            "TableName": "MusicCollection"
        }
    },
    {
        "Get": {
            "Key": {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            "TableName": "MusicCollection"
        }
    }
]
```

Sortie :

```
{
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 4.0,
            "ReadCapacityUnits": 4.0
        }
    ],
    "Responses": [
        {
            "Item": {
                "AlbumTitle": {
```

```
                    "S": "Songs About Life"
                },
                "Artist": {
                    "S": "Acme Band"
                },
                "SongTitle": {
                    "S": "Happy Day"
                }
            }
        },
        {
            "Item": {
                "AlbumTitle": {
                    "S": "Somewhat Famous"
                },
                "Artist": {
                    "S": "No One You Know"
                },
                "SongTitle": {
                    "S": "Call Me Today"
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion de flux de travail complexes avec des transactions DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous TransactGetItemsà la section Référence des AWS CLI commandes.

`transact-write-items`

L'exemple de code suivant montre comment utiliser`transact-write-items`.

AWS CLI

Exemple 1 : écrire des éléments de manière atomique dans une ou plusieurs tables

L'`transact-write-items`exemple suivant met à jour un élément et en supprime un autre. L'opération échoue si l'une des opérations échoue ou si l'un des éléments contient un `Rating` attribut.

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

Contenu du `transact-items.json` fichier :

```
[
    {
        "Update": {
            "Key": {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            "UpdateExpression": "SET AlbumTitle = :newval",
            "ExpressionAttributeValues": {
                ":newval": {"S": "Updated Album Title"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    },
    {
        "Delete": {
            "Key": {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    }
]
```

Sortie :

```
{
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 10.0,
            "WriteCapacityUnits": 10.0
```

```
        }
    ],
    "ItemCollectionMetrics": {
        "MusicCollection": [
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "No One You Know"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            },
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "Acme Band"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Gestion de flux de travail complexes avec des transactions DynamoDB dans le manuel du développeur Amazon DynamoDB.

Exemple 2 : écrire des éléments de manière atomique à l'aide d'un jeton de demande client

La commande suivante utilise un jeton de demande client pour appeler `transact-write-items` idempotent, ce qui signifie que plusieurs appels ont le même effet qu'un seul appel.

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --client-request-token abc123
```

Contenu du `transact-items.json` fichier :

```
[
    {
        "Update": {
            "Key": {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            "UpdateExpression": "SET AlbumTitle = :newval",
            "ExpressionAttributeValues": {
                ":newval": {"S": "Updated Album Title"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    },
    {
        "Delete": {
            "Key": {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    }
]
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de flux de travail complexes avec des transactions DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous TransactWriteItemsà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une balise d'une ressource DynamoDB

L'`untag-resource`exemple suivant supprime de la `MusicCollection` table la balise `Owner` contenant la clé.

```
aws dynamodb untag-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --tag-keys Owner
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage pour DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-continuous-backups`

L'exemple de code suivant montre comment utiliser`update-continuous-backups`.

AWS CLI

Pour mettre à jour les paramètres de sauvegarde continue d'une table DynamoDB

L'`update-continuous-backups`exemple suivant active la point-in-time restauration de la `MusicCollection` table.

```
aws dynamodb update-continuous-backups \
    --table-name MusicCollection \
    --point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

Sortie :

```
{
    "ContinuousBackupsDescription": {
        "ContinuousBackupsStatus": "ENABLED",
        "PointInTimeRecoveryDescription": {
            "PointInTimeRecoveryStatus": "ENABLED",
            "EarliestRestorableDateTime": 1576622404.0,
            "LatestRestorableDateTime": 1576622404.0
        }
    }
}
```

Pour plus d'informations, consultez la section Point-in-Time Recovery for DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateContinuousBackupsà la section Référence des AWS CLI commandes.

`update-contributor-insights`

L'exemple de code suivant montre comment utiliser`update-contributor-insights`.

AWS CLI

Pour activer Contributor Insights sur un tableau

L'`update-contributor-insights`exemple suivant active Contributor Insights sur la `MusicCollection` table et sur l'index secondaire `AlbumTitle-index` global.

```
aws dynamodb update-contributor-insights \
    --table-name MusicCollection \
    --index-name AlbumTitle-index \
    --contributor-insights-action ENABLE
```

Sortie :

```
{
    "TableName": "MusicCollection",
    "IndexName": "AlbumTitle-index",
    "ContributorInsightsStatus": "ENABLING"
}
```

Pour plus d'informations, consultez la section Analyse de l'accès aux données à l'aide de CloudWatch Contributor Insights for DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateContributorInsightsà la section Référence des AWS CLI commandes.

`update-global-table-settings`

L'exemple de code suivant montre comment utiliser`update-global-table-settings`.

AWS CLI

Pour mettre à jour les paramètres de capacité d'écriture provisionnés sur une table globale DynamoDB

L'`update-global-table-settings`exemple suivant définit la capacité d'écriture allouée de la table `MusicCollection` globale à 15.

```
aws dynamodb update-global-table-settings \
    --global-table-name MusicCollection \
    --global-table-provisioned-write-capacity-units 15
```

Sortie :

```
{
    "GlobalTableName": "MusicCollection",
    "ReplicaSettings": [
        {
            "RegionName": "eu-west-1",
            "ReplicaStatus": "UPDATING",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 10,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        },
        {
            "RegionName": "us-east-1",
            "ReplicaStatus": "UPDATING",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 10,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        },
        {
            "RegionName": "us-east-2",
```

```
            "ReplicaStatus": "UPDATING",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 10,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        }
    ]
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateGlobalTableSettingsà la section Référence des AWS CLI commandes.

`update-global-table`

L'exemple de code suivant montre comment utiliser`update-global-table`.

AWS CLI

Pour mettre à jour une table globale DynamoDB

L'`update-global-table`exemple suivant ajoute une réplique dans la région spécifiée à la table `MusicCollection` globale.

```
aws dynamodb update-global-table \
    --global-table-name MusicCollection \
    --replica-updates Create={RegionName=eu-west-1}
```

Sortie :

```
{
    "GlobalTableDescription": {
        "ReplicationGroup": [
            {
                "RegionName": "eu-west-1"
            },
```

```
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
        "CreationDateTime": 1576625818.532,
        "GlobalTableStatus": "ACTIVE",
        "GlobalTableName": "MusicCollection"
    }
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateGlobalTableà la section Référence des AWS CLI commandes.

`update-item`

L'exemple de code suivant montre comment utiliser`update-item`.

AWS CLI

Exemple 1 : pour mettre à jour un élément dans un tableau

L'exemple d'`update-item` suivant met à jour un élément à partir de la table `MusicCollection`. Il ajoute un nouvel attribut (`Year`) et modifie l'`AlbumTitle`attribut. Tous les attributs de l'élément, tels qu'ils apparaissent après la mise à jour, sont renvoyés dans la réponse.

```
aws dynamodb update-item \
    --table-name MusicCollection \
    --key file://key.json \
    --update-expression "SET #Y = :y, #AT = :t" \
    --expression-attribute-names file://expression-attribute-names.json \
    --expression-attribute-values file://expression-attribute-values.json  \
    --return-values ALL_NEW \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

Contenu de `key.json` :

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

Contenu de `expression-attribute-names.json` :

```
{
    "#Y":"Year", "#AT":"AlbumTitle"
}
```

Contenu de `expression-attribute-values.json` :

```
{
    ":y":{"N": "2015"},
    ":t":{"S": "Louder Than Ever"}
}
```

Sortie :

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Louder Than Ever"
        },
        "Awards": {
            "N": "10"
        },
        "Artist": {
            "S": "Acme Band"
        },
        "Year": {
            "N": "2015"
        },
        "SongTitle": {
            "S": "Happy Day"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
```

```
        "CapacityUnits": 3.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
            "Artist": {
                "S": "Acme Band"
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

Pour plus d'informations, consultez la section Rédaction d'un article dans le guide du développeur Amazon DynamoDB.

Exemple 2 : pour mettre à jour un élément de manière conditionnelle

L'exemple suivant met à jour un élément de la `MusicCollection` table, mais uniquement si l'élément existant ne possède pas encore d'`Year`attribut.

```
aws dynamodb update-item \
    --table-name MusicCollection \
    --key file://key.json \
    --update-expression "SET #Y = :y, #AT = :t" \
    --expression-attribute-names file://expression-attribute-names.json \
    --expression-attribute-values file://expression-attribute-values.json  \
    --condition-expression "attribute_not_exists(#Y)"
```

Contenu de `key.json` :

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

Contenu de `expression-attribute-names.json` :

```
{
    "#Y":"Year",
```

```
    "#AT":"AlbumTitle"
}
```

Contenu de `expression-attribute-values.json` :

```
{
    ":y":{"N": "2015"},
    ":t":{"S": "Louder Than Ever"}
}
```

Si l'élément possède déjà un `Year` attribut, DynamoDB renvoie le résultat suivant.

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem
 operation: The conditional request failed
```

Pour plus d'informations, consultez la section Rédaction d'un article dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateItemà la section Référence des AWS CLI commandes.

`update-table-replica-auto-scaling`

L'exemple de code suivant montre comment utiliser`update-table-replica-auto-scaling`.

AWS CLI

Pour mettre à jour les paramètres de mise à l'échelle automatique sur les répliques d'une table globale

L'`update-table-replica-auto-scaling`exemple suivant met à jour les paramètres de mise à l'échelle automatique de la capacité d'écriture sur les répliques de la table globale spécifiée.

```
aws dynamodb update-table-replica-auto-scaling \
    --table-name MusicCollection \
    --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

Contenu de `auto-scaling-policy.json` :

```
{
```

```
    "MinimumUnits": 10,
    "MaximumUnits": 100,
    "AutoScalingDisabled": false,
    "ScalingPolicyUpdate": {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
            "TargetValue": 80
        }
    }
}
```

Sortie :

```
{
    "TableAutoScalingDescription": {
        "TableName": "MusicCollection",
        "TableStatus": "ACTIVE",
        "Replicas": [
            {
                "RegionName": "eu-central-1",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 10,
                    "MaximumUnits": 100,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
```

```
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 80.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            },
            {
                "RegionName": "us-east-1",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 10,
                    "MaximumUnits": 100,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 80.0
                            }
```

```
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            },
            {
                "RegionName": "us-east-2",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 10,
                    "MaximumUnits": 100,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 80.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            }
        ]
    }
```

```
}
```

Pour plus d'informations, consultez les tables globales DynamoDB dans le manuel du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateTableReplicaAutoScalingà la section Référence des AWS CLI commandes.

`update-table`

L'exemple de code suivant montre comment utiliser`update-table`.

AWS CLI

Exemple 1 : pour modifier le mode de facturation d'une table

L'`update-table`exemple suivant augmente la capacité de lecture et d'écriture allouée sur la `MusicCollection` table.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --billing-mode PROVISIONED \
    --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
```

```
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T13:18:18.921000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        }
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une table dans le guide du développeur Amazon DynamoDB.

Exemple 2 : pour créer un index secondaire global

L'exemple suivant ajoute un index secondaire global à la `MusicCollection` table.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \
    --global-secondary-index-updates file://gsi-updates.json
```

Contenu de `gsi-updates.json` :

```
[
    {
        "Create": {
            "IndexName": "AlbumTitle-index",
            "KeySchema": [
                {
                    "AttributeName": "AlbumTitle",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 10,
                "WriteCapacityUnits": 10
            },
            "Projection": {
                "ProjectionType": "ALL"
            }
        }
    }
]
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
```

```
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        },
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitle-index",
                "KeySchema": [
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "HASH"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "CREATING",
                "Backfilling": false,
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 10
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
```

```
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une table dans le guide du développeur Amazon DynamoDB.

Exemple 3 : pour activer DynamoDB Streams sur une table

La commande suivante active DynamoDB Streams sur la table. `MusicCollection`

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
```

```
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        },
        "LocalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "Artist",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "Year",
                        "Genre"
                    ]
                },
                "IndexSizeBytes": 139,
                "ItemCount": 2,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
```

```
            }
        ],
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitle-index",
                "KeySchema": [
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "HASH"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "ACTIVE",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 10
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
            }
        ],
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_IMAGE"
        },
        "LatestStreamLabel": "2020-07-28T21:53:39.112",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une table dans le guide du développeur Amazon DynamoDB.

Exemple 4 : pour activer le chiffrement côté serveur

L'exemple suivant active le chiffrement côté serveur sur la `MusicCollection` table.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --sse-specification Enabled=true,SSEType=KMS
```

Sortie :

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "ACTIVE",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
```

```
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        },
        "LocalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "Artist",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "Year",
                        "Genre"
                    ]
                },
                "IndexSizeBytes": 139,
                "ItemCount": 2,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
            }
        ],
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitle-index",
                "KeySchema": [
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "HASH"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
```

```
                "IndexStatus": "ACTIVE",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 10
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
            }
        ],
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_IMAGE"
        },
        "LatestStreamLabel": "2020-07-28T21:53:39.112",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112",
        "SSEDescription": {
            "Status": "UPDATING"
        }
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une table dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateTableà la section Référence des AWS CLI commandes.

`update-time-to-live`

L'exemple de code suivant montre comment utiliser`update-time-to-live`.

AWS CLI

Pour mettre à jour les paramètres Time to Live sur une table

L'`update-time-to-live`exemple suivant active Time to Live sur la table spécifiée.

```
aws dynamodb update-time-to-live \
    --table-name MusicCollection \
```

```
    --time-to-live-specification Enabled=true,AttributeName=ttl
```

Sortie :

```
{
    "TimeToLiveSpecification": {
        "Enabled": true,
        "AttributeName": "ttl"
    }
}
```

Pour plus d'informations, consultez Time to Live dans le guide du développeur Amazon DynamoDB.

- Pour plus de détails sur l'API, reportez-vous UpdateTimeToLiveà la section Référence des AWS CLI commandes.

Exemples de DynamoDB Streams utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de DynamoDB Streams.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-stream`

L'exemple de code suivant montre comment utiliser`describe-stream`.

AWS CLI

Pour obtenir des informations sur un flux DynamoDB

La `describe-stream` commande suivante affiche des informations sur le flux DynamoDB spécifique.

```
aws dynamodbstreams describe-stream \
    --stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576
```

Sortie :

```
{
    "StreamDescription": {
        "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576",
        "StreamLabel": "2019-10-22T18:02:01.576",
        "StreamStatus": "ENABLED",
        "StreamViewType": "NEW_AND_OLD_IMAGES",
        "CreationRequestDateTime": 1571767321.571,
        "TableName": "Music",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "Shards": [
            {
                "ShardId": "shardId-00000001571767321804-697ce3d2",
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "400000000000642977831",
                    "EndingSequenceNumber": "400000000000642977831"
                }
            },
            {
                "ShardId": "shardId-00000001571780995058-40810d86",
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "757400000000005655171150"
                },
                "ParentShardId": "shardId-00000001571767321804-697ce3d2"
            }
```

```
        ]
    }
}
```

Pour plus d'informations, consultez Capturing Table Activity with DynamoDB Streams dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeStreamà la section Référence des AWS CLI commandes.

`get-records`

L'exemple de code suivant montre comment utiliser`get-records`.

AWS CLI

Pour obtenir des enregistrements à partir d'un flux Dynamodb

La `get-records` commande suivante extrait les enregistrements à l'aide de l'itérateur de partition Amazon DynamoDB spécifié.

```
aws dynamodbstreams get-records \
    --shard-iterator "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmiONmy6xVW4IRcIIbs1zO7MNI
+CjNPlqQjnyRSAnfOwWmKhL1/KNParWSfz2odf780oOObIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QIlOXPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAyvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/lEGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
```

Sortie :

```
{
    "Records": [
        {
            "eventID": "c3b5d798eef6215d42f8137b19a88e50",
            "eventName": "INSERT",
            "eventVersion": "1.1",
            "eventSource": "aws:dynamodb",
            "awsRegion": "us-west-1",
            "dynamodb": {
```

```
                "ApproximateCreationDateTime": 1571849028.0,
                "Keys": {
                    "Artist": {
                        "S": "No One You Know"
                    },
                    "SongTitle": {
                        "S": "Call Me Today"
                    }
                },
                "NewImage": {
                    "AlbumTitle": {
                        "S": "Somewhat Famous"
                    },
                    "Artist": {
                        "S": "No One You Know"
                    },
                    "Awards": {
                        "N": "1"
                    },
                    "SongTitle": {
                        "S": "Call Me Today"
                    }
                },
                "SequenceNumber": "700000000013256296913",
                "SizeBytes": 119,
                "StreamViewType": "NEW_AND_OLD_IMAGES"
            }
        },
        {
            "eventID": "878960a6967867e2da16b27380a27328",
            "eventName": "INSERT",
            "eventVersion": "1.1",
            "eventSource": "aws:dynamodb",
            "awsRegion": "us-west-1",
            "dynamodb": {
                "ApproximateCreationDateTime": 1571849029.0,
                "Keys": {
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
```

```
                "NewImage": {
                    "AlbumTitle": {
                        "S": "Songs About Life"
                    },
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "Awards": {
                        "N": "10"
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
                "SequenceNumber": "800000000013256297217",
                "SizeBytes": 100,
                "StreamViewType": "NEW_AND_OLD_IMAGES"
            }
        },
        {
            "eventID": "520fabde080e159fc3710b15ee1d4daa",
            "eventName": "MODIFY",
            "eventVersion": "1.1",
            "eventSource": "aws:dynamodb",
            "awsRegion": "us-west-1",
            "dynamodb": {
                "ApproximateCreationDateTime": 1571849734.0,
                "Keys": {
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
                "NewImage": {
                    "AlbumTitle": {
                        "S": "Updated Album Title"
                    },
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "Awards": {
                        "N": "10"
```

```
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
                "OldImage": {
                    "AlbumTitle": {
                        "S": "Songs About Life"
                    },
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "Awards": {
                        "N": "10"
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
                "SequenceNumber": "900000000013256687845",
                "SizeBytes": 170,
                "StreamViewType": "NEW_AND_OLD_IMAGES"
            }
        }
    ],
    "NextShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/
Music/stream/2019-10-23T16:41:08.740|1|AAAAAAAAAAEhEI04jkFLW
+LKOwivjT8d/IHEh3iExV2xK00aTxEzVy1C1C7Kbb5+ZOW6bT9VQ2n1/
mrs7+PRiaOZCHJu7JHJVW7zlsqOi/ges3fw8GYEymyL+piEk35cx67rQqwKKyq
+Q6w9JyjreIOj4F2lWLV26lBwRTrIYC4IB7C3BZZK4715QwYdDxNdVHiSBRZX8UqoS6WOt0F87xZLNB9F/
NhYBLXi/wcGvAcBcC0TNIOH+N0NqwtoB/
FGCkNrf8YZ0xRoNN6RgGuVWHF3pxOhxEJeFZoSoJTIKeG9YcYxzi5Ci/
mhdtm7tBXnbw5c6xmsGsBqTirNjlDyJLcWl8Cl0UOLX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0fF/
ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWF2wvwT6Iy2xGrR2r2HzYps9dwGOarVdEITaJfWzNoL4HajMhmREZ
+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PVV6iknY6d19MrdbSTMKoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da
}
```

Pour plus d'informations, consultez Capturing Table Activity with DynamoDB Streams dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetRecordsà la section Référence des AWS CLI commandes.

`get-shard-iterator`

L'exemple de code suivant montre comment utiliser`get-shard-iterator`.

AWS CLI

Pour obtenir un itérateur de partition

La `get-shard-iterator` commande suivante permet de récupérer un itérateur de partition pour la partition spécifiée.

```
aws dynamodbstreams get-shard-iterator \
    --stream-arn arn:aws:dynamodb:us-west-1:12356789012:table/Music/
stream/2019-10-22T18:02:01.576 \
    --shard-id shardId-00000001571780995058-40810d86 \
    --shard-iterator-type LATEST
```

Sortie :

```
{
    "ShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmiONmy6xVW4IRcIIbs1zO7MNI
+CjNPlqQjnyRSAnfOwWmKhL1/KNParWSfz2odf780oOObIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QIlOXPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAyvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/lEGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
}
```

Pour plus d'informations, consultez Capturing Table Activity with DynamoDB Streams dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetShardIteratorà la section Référence des AWS CLI commandes.

`list-streams`

L'exemple de code suivant montre comment utiliser`list-streams`.

AWS CLI

Pour répertorier les flux DynamoDB

La `list-streams` commande suivante répertorie tous les flux Amazon DynamoDB existants dans la région par défaut. AWS

```
aws dynamodbstreams list-streams
```

Sortie :

```
{
    "Streams": [
        {
            "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576",
            "TableName": "Music",
            "StreamLabel": "2019-10-22T18:02:01.576"
        }
    ]
}
```

Pour plus d'informations, consultez Capturing Table Activity with DynamoDB Streams dans le manuel Amazon DynamoDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListStreamsà la section Référence des AWS CLI commandes.

EC2 Exemples Amazon utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon EC2.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-address-transfer`

L'exemple de code suivant montre comment utiliser`accept-address-transfer`.

AWS CLI

Pour accepter le transfert d'une adresse IP Elastic vers votre compte

L'`accept-address-transfer`exemple suivant accepte le transfert de l'adresse IP élastique spécifiée vers votre compte.

```
aws ec2 accept-address-transfer \
    --address 100.21.184.216
```

Sortie :

```
{
    "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "TransferAccountId": "123456789012",
        "TransferOfferExpirationTimestamp": "2023-02-22T20:51:10.000Z",
        "TransferOfferAcceptedTimestamp": "2023-02-22T22:52:54.000Z",
        "AddressTransferStatus": "accepted"
    }
}
```

Pour plus d'informations, consultez Transfer Elastic IP address dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous AcceptAddressTransferà la section Référence des AWS CLI commandes.

`accept-reserved-instances-exchange-quote`

L'exemple de code suivant montre comment utiliser`accept-reserved-instances-exchange-quote`.

AWS CLI

Pour effectuer un échange d'instances réservées convertibles

Cet exemple effectue un échange des instances réservées convertibles spécifiées.

Commande :

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

Sortie :

```
{
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"
}
```

- Pour plus de détails sur l'API, reportez-vous AcceptReservedInstancesExchangeQuoteà la section Référence des AWS CLI commandes.

`accept-transit-gateway-peering-attachment`

L'exemple de code suivant montre comment utiliser`accept-transit-gateway-peering-attachment`.

AWS CLI

Pour accepter une pièce jointe d'appairage d'une passerelle de transit

L'`accept-transit-gateway-peering-attachment`exemple suivant accepte la pièce jointe d'appairage de passerelle de transit spécifiée. Le `--region` paramètre spécifie la région dans laquelle se trouve la passerelle de transit acceptrice.

```
aws ec2 accept-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
    --region us-east-2
```

Sortie :

```
{
```

```
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "pending",
        "CreationTime": "2019-12-09T11:38:31.000Z"
    }
}
```

Pour plus d'informations, consultez la section Transit Gateway peering Attachments dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous AcceptTransitGatewayPeeringAttachmentà la section Référence des AWS CLI commandes.

`accept-transit-gateway-vpc-attachment`

L'exemple de code suivant montre comment utiliser`accept-transit-gateway-vpc-attachment`.

AWS CLI

Accepter une demande d'attachement d'un VPC à une passerelle de transit.

L'`accept-transit-gateway-vpc-attachment`exemple suivant accepte la requête avec la pièce jointe spécifiée.

```
aws ec2 accept-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Sortie :

```
{
    "TransitGatewayVpcAttachment": {
```

```
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "123456789012",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

Pour plus d'informations, voir Transit Gateway Attachments to a VPC dans le Transit Gateways Guide.

- Pour plus de détails sur l'API, reportez-vous AcceptTransitGatewayVpcAttachmentà la section Référence des AWS CLI commandes.

`accept-vpc-endpoint-connections`

L'exemple de code suivant montre comment utiliser`accept-vpc-endpoint-connections`.

AWS CLI

Pour accepter une demande de connexion à un point de terminaison d'interface

Cet exemple accepte la demande de connexion au point de terminaison spécifié pour le service de point de terminaison spécifié.

Commande :

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Sortie :

```
{
  "Unsuccessful": []
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous AcceptVpcEndpointConnectionsà la section Référence des AWS CLI commandes.

`accept-vpc-peering-connection`

L'exemple de code suivant montre comment utiliser`accept-vpc-peering-connection`.

AWS CLI

Pour accepter une connexion d'appairage VPC

Cet exemple accepte la demande de connexion d'appairage VPC spécifiée.

Commande :

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Sortie :

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-44455566",
      "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-1a2b3c4d",
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-111abc45",
      "CidrBlock": "10.0.0.0/28"
    }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous AcceptVpcPeeringConnectionà la section Référence des AWS CLI commandes.

`advertise-byoip-cidr`

L'exemple de code suivant montre comment utiliser`advertise-byoip-cidr`.

AWS CLI

Pour annoncer une plage d'adresses

L'`advertise-byoip-cidr`exemple suivant annonce la plage d' IPv4 adresses publiques spécifiée.

```
aws ec2 advertise-byoip-cidr \
    --cidr 203.0.113.25/24
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
        "State": "provisioned"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous AdvertiseByoipCidrà la section Référence des AWS CLI commandes.

`allocate-address`

L'exemple de code suivant montre comment utiliser`allocate-address`.

AWS CLI

Exemple 1 : allocation d’une adresse IP Elastic à partir d’un groupe d’adresses d’Amazon

L’exemple suivant `allocate-address` alloue une adresse IP Elastic. Amazon EC2 sélectionne l'adresse dans le pool d'adresses d'Amazon.

```
aws ec2 allocate-address
```

Sortie :

```
{
    "PublicIp": "70.224.234.241",
    "AllocationId": "eipalloc-01435ba59eEXAMPLE",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-west-2",
    "Domain": "vpc"
}
```

Pour plus d'informations, consultez la section Adresses IP élastiques dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : allocation d'une adresse IP Elastic et l'associer à un groupe de frontières réseau

L'exemple `allocate-address` suivant alloue une adresse IP Elastic et l'associe au groupe de frontières réseau spécifié.

```
aws ec2 allocate-address \
    --network-border-group us-west-2-lax-1
```

Sortie :

```
{
    "PublicIp": "70.224.234.241",
    "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-west-2-lax-1",
    "Domain": "vpc"
}
```

Pour plus d'informations, consultez la section Adresses IP élastiques dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : allocation d'une adresse IP Elastic à partir de votre propre groupe d'adresses IP

L'exemple `allocate-address` suivant alloue une adresse IP Elastic à partir d'un groupe d'adresses IP que vous avez apporté à votre compte Amazon Web Services. Amazon EC2 sélectionne l'adresse dans le pool d'adresses.

```
aws ec2 allocate-address \
    --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

Sortie :

```
{
    "AllocationId": "eipalloc-02463d08ceEXAMPLE",
    "NetworkBorderGroup": "us-west-2",
    "CustomerOwnedIp": "18.218.95.81",
    "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",
    "Domain": "vpc"
    "NetworkBorderGroup": "us-west-2",
}
```

Pour plus d'informations, consultez la section Adresses IP élastiques dans le guide de EC2 l'utilisateur Amazon.

Exemple 4 : pour allouer une adresse IP élastique à partir d'un pool IPAM

L'`allocate-address`exemple suivant alloue une adresse IP élastique /32 spécifique à partir d'un pool Amazon VPC IP Address Manager (IPAM).

```
aws ec2 allocate-address \
    --region us-east-1 \
    --ipam-pool-id ipam-pool-1234567890abcdef0 \
    --address 192.0.2.0
```

Sortie :

```
{
    "PublicIp": "192.0.2.0",
    "AllocationId": "eipalloc-abcdef01234567890",
    "PublicIpv4Pool": "ipam-pool-1234567890abcdef0",
    "NetworkBorderGroup": "us-east-1",
    "Domain": "vpc"
}
```

Pour plus d'informations, consultez la section Allocation d'adresses IP élastiques séquentielles à partir d'un pool IPAM dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous AllocateAddressà la section Référence des AWS CLI commandes.

`allocate-hosts`

L'exemple de code suivant montre comment utiliser`allocate-hosts`.

AWS CLI

Exemple 1 : pour allouer un hôte dédié

L'`allocate-hosts`exemple suivant alloue un seul hôte dédié dans la zone de `eu-west-1a` disponibilité, sur lequel vous pouvez lancer `m5.large` des instances. Par défaut, l'hôte dédié accepte uniquement les lancements d'instances cibles et ne prend pas en charge la restauration de l'hôte.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --quantity 1
```

Sortie :

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

Exemple 2 : pour allouer un hôte dédié avec le placement automatique et la restauration de l'hôte activés

L'`allocate-hosts`exemple suivant alloue un seul hôte dédié dans la zone de `eu-west-1a` disponibilité avec le placement automatique et la restauration de l'hôte activés.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

Sortie :

```
{
     "HostIds": [
         "h-07879acf49EXAMPLE"
     ]
}
```

Exemple 3 : pour allouer un hôte dédié avec des balises

L'`allocate-hosts`exemple suivant alloue un seul hôte dédié et applique une balise avec une clé nommée `purpose` et une valeur de. `production`

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --quantity 1 \
    --tag-specifications 'ResourceType=dedicated-
host,Tags={Key=purpose,Value=production}'
```

Sortie :

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

Pour plus d'informations, consultez la section Allocation d'un hôte dédié dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous AllocateHostsà la section Référence des AWS CLI commandes.

`allocate-ipam-pool-cidr`

L'exemple de code suivant montre comment utiliser`allocate-ipam-pool-cidr`.

AWS CLI

Pour allouer un CIDR à partir d'un pool IPAM

L'`allocate-ipam-pool-cidr`exemple suivant alloue un CIDR à partir d'un pool IPAM.

(Linux) :

```
aws ec2 allocate-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --netmask-length 24
```

(Fenêtres) :

```
aws ec2 allocate-ipam-pool-cidr ^
   --ipam-pool-id ipam-pool-0533048da7d823723 ^
   --netmask-length 24
```

Sortie :

```
{
    "IpamPoolAllocation": {
        "Cidr": "10.0.0.0/24",
        "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfbd",
        "ResourceType": "custom",
        "ResourceOwner": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Allouer manuellement un CIDR à un pool pour réserver de l'espace d'adresse IP dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous AllocateIpamPoolCidrà la section Référence des AWS CLI commandes.

`apply-security-groups-to-client-vpn-target-network`

L'exemple de code suivant montre comment utiliser`apply-security-groups-to-client-vpn-target-network`.

AWS CLI

Pour appliquer des groupes de sécurité à un réseau cible pour un point de terminaison Client VPN

L'`apply-security-groups-to-client-vpn-target-network`exemple suivant applique le groupe de sécurité `sg-01f6e627a89f4db32` à l'association entre le réseau cible spécifié et le point de terminaison VPN du Client.

```
aws ec2 apply-security-groups-to-client-vpn-target-network \
    --security-group-ids sg-01f6e627a89f4db32 \
    --vpc-id vpc-0e2110c2f324332e0 \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "SecurityGroupIds": [
        "sg-01f6e627a89f4db32"
    ]
}
```

Pour plus d'informations, consultez la section Réseaux cibles dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous ApplySecurityGroupsToClientVpnTargetNetworkà la section Référence des AWS CLI commandes.

`assign-ipv6-addresses`

L'exemple de code suivant montre comment utiliser`assign-ipv6-addresses`.

AWS CLI

Pour attribuer des IPv6 adresses spécifiques à une interface réseau

Cet exemple attribue les IPv6 adresses spécifiées à l'interface réseau spécifiée.

Commande :

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 2001:db8:1234:1a00:9691:9503:25ad:1761
```

Sortie :

```
{
  "AssignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071",
      "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
```

```
}
```

Pour attribuer IPv6 les adresses sélectionnées par Amazon à une interface réseau

Cet exemple attribue deux IPv6 adresses à l'interface réseau spécifiée. Amazon attribue automatiquement ces IPv6 adresses à partir des IPv6 adresses disponibles dans la plage de blocs IPv6 CIDR du sous-réseau.

Commande :

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-
count 2
```

Sortie :

```
{
  "AssignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071",
      "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

- Pour plus de détails sur l'API, consultez la section AssignIpv6Addresses dans AWS CLI Command Reference.

`assign-private-ip-addresses`

L'exemple de code suivant montre comment utiliser`assign-private-ip-addresses`.

AWS CLI

Pour attribuer une interface réseau à une adresse IP privée secondaire spécifique

Cet exemple attribue l'adresse IP privée secondaire spécifiée à l'interface réseau spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-
ip-addresses 10.0.0.82
```

Pour attribuer des adresses IP privées secondaires sélectionnées par EC2 Amazon à une interface réseau

Cet exemple attribue deux adresses IP privées secondaires à l'interface réseau spécifiée. Amazon attribue EC2 automatiquement ces adresses IP à partir des adresses IP disponibles dans la plage de blocs CIDR du sous-réseau auquel l'interface réseau est associée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-private-ip-address-count 2
```

- Pour plus de détails sur l'API, reportez-vous AssignPrivateIpAddressesà la section Référence des AWS CLI commandes.

`assign-private-nat-gateway-address`

L'exemple de code suivant montre comment utiliser`assign-private-nat-gateway-address`.

AWS CLI

Pour attribuer des adresses IP privées à votre passerelle NAT privée

L'`assign-private-nat-gateway-address`exemple suivant attribue deux adresses IP privées à la passerelle NAT privée spécifiée.

```
aws ec2 assign-private-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --private-ip-address-count 2
```

Sortie :

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "IsPrimary": false,
            "Status": "assigning"
        },
```

```
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "IsPrimary": false,
            "Status": "assigning"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous AssignPrivateNatGatewayAddressà la section Référence des AWS CLI commandes.

`associate-address`

L'exemple de code suivant montre comment utiliser`associate-address`.

AWS CLI

Exemple 1 : pour associer une adresse IP élastique à une instance

L'`associate-address`exemple suivant associe une adresse IP élastique à l' EC2 instance spécifiée.

```
aws ec2 associate-address \
    --instance-id i-0b263919b6498b123 \
    --allocation-id eipalloc-64d5890a
```

Sortie :

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

Exemple 2 : pour associer une adresse IP élastique à une interface réseau

L'`associate-address`exemple suivant associe l'adresse IP élastique spécifiée à l'interface réseau spécifiée.

```
aws ec2 associate-address
```

```
    --allocation-id eipalloc-64d5890a \
    --network-interface-id eni-1a2b3c4d
```

Sortie :

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

Exemple 3 : pour associer une adresse IP élastique à une adresse IP privée

L'`associate-address`exemple suivant associe l'adresse IP élastique spécifiée à l'adresse IP privée spécifiée dans l'interface réseau spécifiée.

```
aws ec2 associate-address \
    --allocation-id eipalloc-64d5890a \
    --network-interface-id eni-1a2b3c4d \
    --private-ip-address 10.0.0.85
```

Sortie :

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

Pour plus d'informations, consultez la section Adresses IP élastiques dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous AssociateAddressà la section Référence des AWS CLI commandes.

`associate-client-vpn-target-network`

L'exemple de code suivant montre comment utiliser`associate-client-vpn-target-network`.

AWS CLI

Pour associer un réseau cible à un point de terminaison VPN du Client

L'`associate-client-vpn-target-network`exemple suivant associe un sous-réseau au point de terminaison Client VPN spécifié.

```
aws ec2 associate-client-vpn-target-network \
    --subnet-id subnet-0123456789abcabca \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "AssociationId": "cvpn-assoc-12312312312312312",
    "Status": {
        "Code": "associating"
    }
}
```

Pour plus d'informations, consultez la section Réseaux cibles dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous AssociateClientVpnTargetNetworkà la section Référence des AWS CLI commandes.

`associate-dhcp-options`

L'exemple de code suivant montre comment utiliser`associate-dhcp-options`.

AWS CLI

Pour associer un ensemble d'options DHCP à votre VPC

Cet exemple associe le jeu d'options DHCP spécifié au VPC spécifié. Si la commande réussit, aucune sortie n’est renvoyée.

Commande :

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

Pour associer les options DHCP définies par défaut à votre VPC

Cet exemple associe les options DHCP définies par défaut au VPC spécifié. Si la commande réussit, aucune sortie n’est renvoyée.

Commande :

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- Pour plus de détails sur l'API, reportez-vous AssociateDhcpOptionsà la section Référence des AWS CLI commandes.

`associate-iam-instance-profile`

L'exemple de code suivant montre comment utiliser`associate-iam-instance-profile`.

AWS CLI

Pour associer un profil d'instance IAM à une instance

Cet exemple associe un profil d'instance IAM nommé `admin-role` à instance`i-123456789abcde123`.

Commande :

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-instance-profile Name=admin-role
```

Sortie :

```
{
  "IamInstanceProfileAssociation": {
      "InstanceId": "i-123456789abcde123",
      "State": "associating",
      "AssociationId": "iip-assoc-0e7736511a163c209",
      "IamInstanceProfile": {
          "Id": "AIPAJBLK7RKJKWDXVHIEC",
          "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateIamInstanceProfileà la section Référence des AWS CLI commandes.

`associate-instance-event-window`

L'exemple de code suivant montre comment utiliser`associate-instance-event-window`.

AWS CLI

Exemple 1 : pour associer une ou plusieurs instances à une fenêtre d'événements

L'`associate-instance-event-window`exemple suivant associe une ou plusieurs instances à une fenêtre d'événements.

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour associer des balises d'instance à une fenêtre d'événements

L'`associate-instance-event-window`exemple suivant associe des balises d'instance à une fenêtre d'événements. Entrez un `instance-event-window-id` paramètre pour définir la fenêtre d'événements. Pour associer des balises d'instance, spécifiez le `association-target` paramètre, et pour la valeur du paramètre, spécifiez une ou plusieurs balises.

```
aws ec2 associate-instance-event-window \
```

```
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [
                {
                    "Key": "k2",
                    "Value": "v2"
                },
                {
                    "Key": "k1",
                    "Value": "v1"
                }
            ],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour associer un hôte dédié à une fenêtre d'événements

L'`associate-instance-event-window`exemple suivant associe un hôte dédié à une fenêtre d'événements. Entrez un `instance-event-window-id` paramètre pour définir la fenêtre d'événements. Pour associer un hôte dédié, spécifiez le `--association-target` paramètre, et pour les valeurs des paramètres, spécifiez un ou plusieurs hôtes dédiés IDs.

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
```

```
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
            ]
        },
        "State": "creating"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous AssociateInstanceEventWindowà la section Référence des AWS CLI commandes.

`associate-ipam-resource-discovery`

L'exemple de code suivant montre comment utiliser`associate-ipam-resource-discovery`.

AWS CLI

Pour associer une découverte de ressources à un IPAM

Dans cet exemple, vous êtes un administrateur délégué IPAM et une découverte de ressources a été créée et partagée avec vous par un autre AWS compte afin que vous puissiez utiliser IPAM pour gérer et surveiller les ressources CIDRs détenues par l'autre compte.

Remarque

Pour terminer cette demande, vous aurez besoin de l'ID de découverte de ressources que vous pouvez obtenir describe-ipam-resource-discoveriesset de l'ID IPAM que vous pouvez obtenir

avec describe-ipams. La découverte de la ressource que vous associez doit d'abord avoir été partagée avec votre compte via la AWS RAM. `--region` Vous entrez doit correspondre à la région d'origine de l'IPAM à laquelle vous l'associez.

L'`associate-ipam-resource-discovery`exemple suivant associe une découverte de ressources à un IPAM.

```
aws ec2 associate-ipam-resource-discovery \
    --ipam-id ipam-005f921c17ebd5107 \
    --ipam-resource-discovery-id ipam-res-disco-03e0406de76a044ee \
    --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-
center,Value=cc123}]' \
    --region us-east-1
```

Sortie :

```
{
    {
        "IpamResourceDiscoveryAssociation": {
            "OwnerId": "320805250157",
            "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
            "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-04382a6346357cf82",
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "IpamId": "ipam-005f921c17ebd5107",
            "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
            "IpamRegion": "us-east-1",
            "IsDefault": false,
            "ResourceDiscoveryStatus": "active",
            "State": "associate-in-progress",
            "Tags": []
        }
    }
}
```

Une fois que vous avez associé une découverte de ressources, vous pouvez surveiller et and/or gérer les adresses IP des ressources créées par les autres comptes. Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous AssociateIpamResourceDiscoveryà la section Référence des AWS CLI commandes.

`associate-nat-gateway-address`

L'exemple de code suivant montre comment utiliser`associate-nat-gateway-address`.

AWS CLI

Pour associer une adresse IP élastique à une passerelle NAT publique

L'`associate-nat-gateway-address`exemple suivant associe l'adresse IP élastique spécifiée à la passerelle NAT publique spécifiée. AWS attribue automatiquement une IPv4 adresse privée secondaire.

```
aws ec2 associate-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --allocation-ids eipalloc-0be6ecac95EXAMPLE
```

Sortie :

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
            "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
            "IsPrimary": false,
            "Status": "associating"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous AssociateNatGatewayAddressà la section Référence des AWS CLI commandes.

`associate-route-table`

L'exemple de code suivant montre comment utiliser`associate-route-table`.

AWS CLI

Pour associer une table de routage à un sous-réseau

Cet exemple associe la table de routage spécifiée au sous-réseau spécifié.

Commande :

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-
id subnet-9d4a7b6c
```

Sortie :

```
{
    "AssociationId": "rtbassoc-781d0d1a"
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateRouteTableà la section Référence des AWS CLI commandes.

`associate-security-group-vpc`

L'exemple de code suivant montre comment utiliser`associate-security-group-vpc`.

AWS CLI

Pour associer un groupe de sécurité à un autre VPC

L'`associate-security-group-vpc`exemple suivant associe le groupe de sécurité spécifié au VPC spécifié.

```
aws ec2 associate-security-group-vpc \
    --group-id sg-04dbb43907d3f8a78 \
    --vpc-id vpc-0bf4c2739bc05a694
```

Sortie :

```
{
    "State": "associating"
}
```

Pour plus d'informations, consultez Associer des groupes de sécurité à plusieurs VPCs dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous AssociateSecurityGroupVpcà la section Référence des AWS CLI commandes.

`associate-subnet-cidr-block`

L'exemple de code suivant montre comment utiliser`associate-subnet-cidr-block`.

AWS CLI

Pour associer un bloc IPv6 CIDR à un sous-réseau

Cet exemple associe un bloc IPv6 CIDR au sous-réseau spécifié.

Commande :

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-
block 2001:db8:1234:1a00::/64
```

Sortie :

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
      "Ipv6CidrBlockState": {
          "State": "associating"
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateSubnetCidrBlockà la section Référence des AWS CLI commandes.

`associate-transit-gateway-multicast-domain`

L'exemple de code suivant montre comment utiliser`associate-transit-gateway-multicast-domain`.

AWS CLI

Pour associer une passerelle de transit à un domaine de multidiffusion

L'`associate-transit-gateway-multicast-domain`exemple suivant associe le sous-réseau et la pièce jointe spécifiés au domaine de multidiffusion spécifié.

```
aws ec2 associate-transit-gateway-multicast-domain \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --transit-gateway-attachment-id tgw-attach-028c1dd0f8f5cbe8e \
    --subnet-ids subnet-000de86e3b49c932a \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Sortie :

```
{
    "Associations": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",
        "ResourceId": "vpc-01128d2c240c09bd5",
        "ResourceType": "vpc",
        "Subnets": [
            {
                "SubnetId": "subnet-000de86e3b49c932a",
                "State": "associating"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Domaines de multidiffusion dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous AssociateTransitGatewayMulticastDomainà la section Référence des AWS CLI commandes.

`associate-transit-gateway-route-table`

L'exemple de code suivant montre comment utiliser`associate-transit-gateway-route-table`.

AWS CLI

Pour associer une table de routage de passerelle de transit à une pièce jointe de passerelle de transit

L'exemple suivant associe la table de routage de passerelle de transit spécifiée à la pièce jointe VPC spécifiée.

```
aws ec2 associate-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Sortie :

```
{
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "ResourceId": "vpc-0065acced4EXAMPLE",
        "ResourceType": "vpc",
        "State": "associating"
    }
}
```

Pour plus d'informations, consultez la section Associer une table de routage de Transit Gateway dans le Guide des passerelles de AWS transit.

- Pour plus de détails sur l'API, reportez-vous AssociateTransitGatewayRouteTableà la section Référence des AWS CLI commandes.

`associate-vpc-cidr-block`

L'exemple de code suivant montre comment utiliser`associate-vpc-cidr-block`.

AWS CLI

Exemple 1 : pour associer un bloc IPv6 CIDR fourni par Amazon à un VPC

L'`associate-vpc-cidr-block`exemple suivant associe un bloc IPv6 CIDR au VPC spécifié. :

```
aws ec2 associate-vpc-cidr-block \
    --amazon-provided-ipv6-cidr-block \
    --ipv6-cidr-block-network-border-group us-west-2-lax-1  \
    --vpc-id vpc-8EXAMPLE
```

Sortie :

```
{
    "Ipv6CidrBlockAssociation": {
        "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",
        "Ipv6CidrBlockState": {
            "State": "associating"
        },
        "NetworkBorderGroup": "us-west-2-lax-1"
    },
    "VpcId": "vpc-8EXAMPLE"
}
```

Exemple 2 : Pour associer un bloc IPv4 CIDR supplémentaire à un VPC

L'`associate-vpc-cidr-block`exemple suivant associe le bloc IPv4 CIDR `10.2.0.0/16` au VPC spécifié.

```
aws ec2 associate-vpc-cidr-block \
    --vpc-id vpc-1EXAMPLE \
    --cidr-block 10.2.0.0/16
```

Sortie :

```
{
    "CidrBlockAssociation": {
        "AssociationId": "vpc-cidr-assoc-2EXAMPLE",
        "CidrBlock": "10.2.0.0/16",
        "CidrBlockState": {
            "State": "associating"
        }
    },
    "VpcId": "vpc-1EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateVpcCidrBlockà la section Référence des AWS CLI commandes.

`attach-classic-link-vpc`

L'exemple de code suivant montre comment utiliser`attach-classic-link-vpc`.

AWS CLI

Pour lier (attacher) une instance EC2 -Classic à un VPC

Cet exemple lie l'instance i-1234567890abcdef0 au VPC vpc-88888888 via le groupe de sécurité VPC sg-12312312.

Commande :

```
aws ec2 attach-classic-link-vpc --instance-id  i-1234567890abcdef0 --vpc-
id vpc-88888888 --groups sg-12312312
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous AttachClassicLinkVpcà la section Référence des AWS CLI commandes.

`attach-internet-gateway`

L'exemple de code suivant montre comment utiliser`attach-internet-gateway`.

AWS CLI

Pour associer une passerelle Internet à votre VPC

L'`attach-internet-gateway`exemple suivant attache la passerelle Internet spécifiée au VPC spécifique.

```
aws ec2 attach-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE \
```

```
    --vpc-id vpc-0a60eb65b4EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Passerelles Internet dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous AttachInternetGatewayà la section Référence des AWS CLI commandes.

`attach-network-interface`

L'exemple de code suivant montre comment utiliser`attach-network-interface`.

AWS CLI

Exemple 1 : pour associer une interface réseau à une instance

L'`attach-network-interface`exemple suivant attache l'interface réseau spécifiée à l'instance spécifiée.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-0dc56a8d4640ad10a \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

Sortie :

```
{
    "AttachmentId": "eni-attach-01a8fc87363f07cf9"
}
```

Pour plus d'informations, consultez la section Elastic network interfaces dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour associer une interface réseau à une instance comportant plusieurs cartes réseau

L'`attach-network-interface`exemple suivant attache l'interface réseau spécifiée à l'instance et à la carte réseau spécifiées.

```
aws ec2 attach-network-interface \
```

```
    --network-interface-id eni-07483b1897541ad83 \
    --instance-id i-01234567890abcdef \
    --network-card-index 1 \
    --device-index 1
```

Sortie :

```
{
    "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"
}
```

Pour plus d'informations, consultez la section Elastic network interfaces dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous AttachNetworkInterfaceà la section Référence des AWS CLI commandes.

`attach-verified-access-trust-provider`

L'exemple de code suivant montre comment utiliser`attach-verified-access-trust-provider`.

AWS CLI

Pour associer un fournisseur de confiance à une instance

L'`attach-verified-access-trust-provider`exemple suivant attache le fournisseur de confiance Verified Access spécifié à l'instance Verified Access spécifiée.

```
aws ec2 attach-verified-access-trust-provider \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Sortie :

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
```

```
        "CreationTime": "2023-08-25T19:00:38",
        "LastUpdatedTime": "2023-08-25T19:00:38"
    },
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "",
        "VerifiedAccessTrustProviders": [
            {
                "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
                "TrustProviderType": "user",
                "UserTrustProviderType": "iam-identity-center"
            }
        ],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T18:27:56"
    }
}
```

Pour plus d'informations, consultez la section Instances d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous AttachVerifiedAccessTrustProviderà la section Référence des AWS CLI commandes.

`attach-volume`

L'exemple de code suivant montre comment utiliser`attach-volume`.

AWS CLI

Pour attacher un volume à une instance

Cet exemple de commande attache un volume (`vol-1234567890abcdef0`) à une instance (`i-01474ef662b89480`) en tant que`/dev/sdf`.

Commande :

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-
id i-01474ef662b89480 --device /dev/sdf
```

Sortie :

```
{
```

```
    "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "InstanceId": "i-01474ef662b89480",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "attaching",
    "Device": "/dev/sdf"
}
```

- Pour plus de détails sur l'API, reportez-vous AttachVolumeà la section Référence des AWS CLI commandes.

`attach-vpn-gateway`

L'exemple de code suivant montre comment utiliser`attach-vpn-gateway`.

AWS CLI

Pour associer une passerelle privée virtuelle à votre VPC

L'`attach-vpn-gateway`exemple suivant attache la passerelle privée virtuelle spécifiée au VPC spécifié.

```
aws ec2 attach-vpn-gateway \
    --vpn-gateway-id vgw-9a4cacf3 \
    --vpc-id vpc-a01106c2
```

Sortie :

```
{
    "VpcAttachment": {
        "State": "attaching",
        "VpcId": "vpc-a01106c2"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous AttachVpnGatewayà la section Référence des AWS CLI commandes.

`authorize-client-vpn-ingress`

L'exemple de code suivant montre comment utiliser`authorize-client-vpn-ingress`.

AWS CLI

Pour ajouter une règle d'autorisation pour un point de terminaison VPN client

L'`authorize-client-vpn-ingress`exemple suivant ajoute une règle d'autorisation d'entrée qui permet à tous les clients d'accéder à Internet (`0.0.0.0/0`).

```
aws ec2 authorize-client-vpn-ingress \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --target-network-cidr 0.0.0.0/0 \
    --authorize-all-groups
```

Sortie :

```
{
    "Status": {
        "Code": "authorizing"
    }
}
```

Pour plus d'informations, consultez la section Règles d'autorisation dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous AuthorizeClientVpnIngressà la section Référence des AWS CLI commandes.

`authorize-security-group-egress`

L'exemple de code suivant montre comment utiliser`authorize-security-group-egress`.

AWS CLI

Exemple 1 : pour ajouter une règle autorisant le trafic sortant vers une plage d'adresses spécifique

L'`authorize-security-group-egress`exemple suivant ajoute une règle qui autorise l'accès aux plages d'adresses spécifiées sur le port TCP 80.

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b15794cdb17bf29c",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "10.0.0.0/16"
        }
    ]
}
```

Exemple 2 : pour ajouter une règle autorisant le trafic sortant vers un groupe de sécurité spécifique

L'`authorize-security-group-egress`exemple suivant ajoute une règle qui accorde l'accès au groupe de sécurité spécifié sur le port TCP 80.

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bbeec5c22}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b5dd815afcea9cc3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
```

```
            "ReferencedGroupInfo": {
                "GroupId": "sg-0aad1c26bbeec5c22",
                "UserId": "123456789012"
            }
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter Groupes de sécurité dans le Guide de l'utilisateur Amazon VPC (français non garanti).

- Pour plus de détails sur l'API, reportez-vous AuthorizeSecurityGroupEgressà la section Référence des AWS CLI commandes.

`authorize-security-group-ingress`

L'exemple de code suivant montre comment utiliser`authorize-security-group-ingress`.

AWS CLI

Exemple 1 : ajout d'une règle autorisant le trafic SSH entrant

L'exemple `authorize-security-group-ingress` suivant ajoute une règle qui autorise le trafic entrant sur le port TCP 22 (SSH).

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-01afa97ef3e1bedfc",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
```

```
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "203.0.113.0/24"
        }
    ]
}
```

Exemple 2 : ajout d'une règle autorisant le trafic HTTP entrant en provenance d'un autre groupe de sécurité

L'exemple `authorize-security-group-ingress` suivant ajoute une règle qui autorise l'accès entrant sur le port TCP 80 à partir du groupe de sécurité source `sg-1a2b3c4d`. Le groupe source doit se trouver dans le même VPC ou dans un VPC homologue (nécessite une connexion d'appairage de VPC). Le trafic entrant est autorisé en fonction des adresses IP privées des instances associées au groupe de sécurité source (et non de l'adresse IP publique ou de l'adresse IP Elastic).

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp \
    --port 80 \
    --source-group sg-1a2b3c4d
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-01f4be99110f638a7",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "ReferencedGroupInfo": {
                "GroupId": "sg-1a2b3c4d",
                "UserId": "123456789012"
            }
        }
```

```
    ]
}
```

Exemple 3 : ajout de plusieurs règles dans le même appel

L'exemple `authorize-security-group-ingress` suivant utilise le paramètre `ip-permissions` pour ajouter deux règles entrantes, l'une qui active l'accès entrant sur le port TCP 3389 (RDP) et l'autre qui active le ping/ICMP.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges=[{CidrIp=172.31.0.0/16}]"
 "IpProtocol=icmp,FromPort=-1,ToPort=-1,IpRanges=[{CidrIp=172.31.0.0/16}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-00e06e5d3690f29f3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "172.31.0.0/16"
        },
        {
            "SecurityGroupRuleId": "sgr-0a133dd4493944b87",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "172.31.0.0/16"
        }
    ]
}
```

Exemple 4 : ajout d'une règle pour le trafic ICMP

L'exemple `authorize-security-group-ingress` suivant utilise le paramètre `ip-permissions` pour ajouter une règle entrante qui autorise le message ICMP `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (type 3, code 4) depuis n'importe où.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0de3811019069b787",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "icmp",
            "FromPort": 3,
            "ToPort": 4,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

Exemple 5 : pour ajouter une règle pour le IPv6 trafic

L'`authorize-security-group-ingress`exemple suivant utilise le `ip-permissions` paramètre pour ajouter une règle entrante qui autorise l'accès SSH (port 22) à partir de la IPv6 plage. `2001:db8:1234:1a00::/64`

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=22,ToPort=22,Ipv6Ranges=[{CidrIpv6=2001:db8:1234:1a00::/64}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0455bc68b60805563",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv6": "2001:db8:1234:1a00::/64"
        }
    ]
}
```

Exemple 6 : pour ajouter une règle pour le ICMPv6 trafic

L'`authorize-security-group-ingress`exemple suivant utilise le `ip-permissions` paramètre pour ajouter une règle entrante qui autorise le ICMPv6 trafic en provenance de n'importe où.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions 'IpProtocol=icmpv6,Ipv6Ranges=[{CidrIpv6=::/0}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "icmpv6",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv6": "::/0"
```

```
        }
    ]
}
```

Exemple 7 : ajout d'une règle avec une description

L'`authorize-security-group-ingress`exemple suivant utilise le `ip-permissions` paramètre pour ajouter une règle entrante qui autorise le trafic RDP depuis la plage d' IPv4 adresses spécifiée. La règle comprend une description qui vous aidera à l'identifier ultérieurement.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges=[{CidrIp=203.0.113.0/24,Description='RDP
 office'}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0397bbcc01e974db3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "203.0.113.0/24",
            "Description": "RDP access from NY office"
        }
    ]
}
```

Exemple 8 : ajout d'une règle entrante utilisant une liste de préfixes

L'exemple `authorize-security-group-ingress` suivant utilise le paramètre `ip-permissions` pour ajouter une règle entrante qui autorise tout le trafic pour les plages CIDR de la liste de préfixes spécifiée.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-04a351bfe432d4e71 \
    --ip-permissions
 'IpProtocol=all,PrefixListIds=[{PrefixListId=pl-002dc3ec097de1514}]'
```

Sortie :

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "PrefixListId": "pl-0721453c7ac4ec009"
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter Groupes de sécurité dans le Guide de l'utilisateur Amazon VPC (français non garanti).

- Pour plus de détails sur l'API, reportez-vous AuthorizeSecurityGroupIngressà la section Référence des AWS CLI commandes.

`bundle-instance`

L'exemple de code suivant montre comment utiliser`bundle-instance`.

AWS CLI

Pour regrouper une instance

Cet exemple regroupe l'instance `i-1234567890abcdef0` dans un compartiment appelé`bundletasks`. Avant de spécifier des valeurs pour votre clé d'accès IDs, consultez et suivez les instructions de la section Meilleures pratiques de gestion des clés AWS d'accès.

Commande :

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

Sortie :

```
{
  "BundleTask": {
    "UpdateTime": "2015-09-15T13:30:35.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "Storage": {
      "S3": {
        "Prefix": "winami",
        "Bucket": "bundletasks"
      }
    },
    "State": "pending",
    "StartTime": "2015-09-15T13:30:35.000Z",
    "BundleId": "bun-294e041f"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous BundleInstanceà la section Référence des AWS CLI commandes.

`cancel-bundle-task`

L'exemple de code suivant montre comment utiliser`cancel-bundle-task`.

AWS CLI

Pour annuler une tâche groupée

Cet exemple annule une tâche groupée. `bun-2a4e041c`

Commande :

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

Sortie :

```
{
  "BundleTask": {
```

```
    "UpdateTime": "2015-09-15T13:27:40.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "Storage": {
      "S3": {
        "Prefix": "winami",
        "Bucket": "bundletasks"
      }
    },
    "State": "cancelling",
    "StartTime": "2015-09-15T13:24:35.000Z",
    "BundleId": "bun-2a4e041c"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CancelBundleTaskà la section Référence des AWS CLI commandes.

`cancel-capacity-reservation-fleets`

L'exemple de code suivant montre comment utiliser`cancel-capacity-reservation-fleets`.

AWS CLI

Pour annuler une flotte de réservation de capacité

L'`cancel-capacity-reservation-fleets`exemple suivant annule la flotte de réservation de capacité spécifiée et la capacité qu'elle réserve. Lorsque vous annulez une flotte, son statut passe à `cancelled` et elle ne peut plus créer de nouvelles réservations de capacité. En outre, toutes les réservations de capacité individuelles de la flotte sont annulées, et les instances qui fonctionnaient auparavant dans la capacité réservée continuent de fonctionner normalement en capacité partagée.

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

Sortie :

```
{
    "SuccessfulFleetCancellations": [
        {
            "CurrentFleetState": "cancelling",
```

```
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
}
```

Pour plus d'informations sur les flottes de réservation de capacité, consultez la section Flottes de réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CancelCapacityReservationFleetsà la section Référence des AWS CLI commandes.

`cancel-capacity-reservation`

L'exemple de code suivant montre comment utiliser`cancel-capacity-reservation`.

AWS CLI

Pour annuler une réservation de capacité

L'`cancel-capacity-reservation`exemple suivant annule la réservation de capacité spécifiée.

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez Annuler une réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CancelCapacityReservationà la section Référence des AWS CLI commandes.

`cancel-conversion-task`

L'exemple de code suivant montre comment utiliser`cancel-conversion-task`.

AWS CLI

Pour annuler la conversion active d'une instance ou d'un volume

Cet exemple annule le téléchargement associé à l'ID de tâche import-i-fh 95npoc. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- Pour plus de détails sur l'API, reportez-vous CancelConversionTaskà la section Référence des AWS CLI commandes.

`cancel-export-task`

L'exemple de code suivant montre comment utiliser`cancel-export-task`.

AWS CLI

Pour annuler une tâche d'exportation active

Cet exemple annule une tâche d'exportation active portant l'ID de tâche export-i-fgelt 0i7. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- Pour plus de détails sur l'API, reportez-vous CancelExportTaskà la section Référence des AWS CLI commandes.

`cancel-image-launch-permission`

L'exemple de code suivant montre comment utiliser`cancel-image-launch-permission`.

AWS CLI

Pour annuler le partage d'une AMI avec votre compte Amazon Web Services

L'`cancel-image-launch-permission`exemple suivant supprime votre compte des autorisations de lancement de l'AMI spécifiée.

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0123456789example \
    --region us-east-1
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez la section Annuler le partage d'une AMI avec votre compte Amazon Web Services dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CancelImageLaunchPermissionà la section Référence des AWS CLI commandes.

`cancel-import-task`

L'exemple de code suivant montre comment utiliser`cancel-import-task`.

AWS CLI

Pour annuler une tâche d'importation

L'`cancel-import-task`exemple suivant annule la tâche d'importation d'image spécifiée.

```
aws ec2 cancel-import-task \
    --import-task-id import-ami-1234567890abcdef0
```

Sortie :

```
{
    "ImportTaskId": "import-ami-1234567890abcdef0",
    "PreviousState": "active",
    "State": "deleting"
}
```

- Pour plus de détails sur l'API, reportez-vous CancelImportTaskà la section Référence des AWS CLI commandes.

`cancel-reserved-instances-listing`

L'exemple de code suivant montre comment utiliser`cancel-reserved-instances-listing`.

AWS CLI

Pour annuler une liste d'instances réservées

L'`cancel-reserved-instances-listing`exemple suivant annule la liste des instances réservées spécifiée.

```
aws ec2 cancel-reserved-instances-listing \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- Pour plus de détails sur l'API, reportez-vous CancelReservedInstancesListingà la section Référence des AWS CLI commandes.

`cancel-spot-fleet-requests`

L'exemple de code suivant montre comment utiliser`cancel-spot-fleet-requests`.

AWS CLI

Exemple 1 : pour annuler une demande de flotte Spot et mettre fin aux instances associées

L'`cancel-spot-fleet-requests`exemple suivant annule une demande de parc Spot et met fin aux instances à la demande et aux instances ponctuelles associées.

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

Sortie :

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
```

```
    ],
    "UnsuccessfulFleetRequests": []
}
```

Exemple 2 : Pour annuler une demande de flotte Spot sans mettre fin aux instances associées

L'`cancel-spot-fleet-requests`exemple suivant annule une demande de parc Spot sans mettre fin aux instances à la demande et aux instances ponctuelles associées.

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

Sortie :

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

Pour plus d'informations, consultez la section Annuler une demande de Spot Fleet dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CancelSpotFleetRequestsà la section Référence des AWS CLI commandes.

`cancel-spot-instance-requests`

L'exemple de code suivant montre comment utiliser`cancel-spot-instance-requests`.

AWS CLI

Pour annuler les demandes d'instance Spot

Cet exemple de commande annule une demande d'instance Spot.

Commande :

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Sortie :

```
{
    "CancelledSpotInstanceRequests": [
        {
            "State": "cancelled",
            "SpotInstanceRequestId": "sir-08b93456"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CancelSpotInstanceRequestsà la section Référence des AWS CLI commandes.

`confirm-product-instance`

L'exemple de code suivant montre comment utiliser`confirm-product-instance`.

AWS CLI

Pour confirmer l'instance du produit

Cet exemple détermine si le code produit spécifié est associé à l'instance spécifiée.

Commande :

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-
id i-1234567890abcdef0
```

Sortie :

```
{
  "OwnerId": "123456789012"
}
```

- Pour plus de détails sur l'API, reportez-vous ConfirmProductInstanceà la section Référence des AWS CLI commandes.

`copy-fpga-image`

L'exemple de code suivant montre comment utiliser`copy-fpga-image`.

AWS CLI

Pour copier une image Amazon FPGA

Cet exemple copie l'AFI spécifié de la `us-east-1` région vers la région actuelle (`eu-west-1`).

Commande :

```
aws ec2 copy-fpga-image --name copy-afi --source-fpga-image-id afi-0d123e123bfc85abc
 --source-region us-east-1 --region eu-west-1
```

Sortie :

```
{
  "FpgaImageId": "afi-06b12350a123fbabc"
}
```

- Pour plus de détails sur l'API, reportez-vous CopyFpgaImageà la section Référence des AWS CLI commandes.

`copy-image`

L'exemple de code suivant montre comment utiliser`copy-image`.

AWS CLI

Exemple 1 : pour copier une AMI dans une autre région

L'`copy-image`exemple de commande suivant copie l'AMI spécifiée de la `us-west-2` région vers la `us-east-1` région et ajoute une brève description.

```
aws ec2 copy-image \
    --region us-east-1 \
    --name ami-name \
    --source-region us-west-2 \
    --source-image-id ami-066877671789bd71b \
    --description "This is my copied image."
```

Sortie :

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

Pour plus d'informations, consultez Copier une AMI dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour copier une AMI dans une autre région et chiffrer l'instantané de sauvegarde

La `copy-image` commande suivante copie l'AMI spécifiée de la `us-west-2` région vers la région actuelle et chiffre le snapshot de sauvegarde à l'aide de la clé KMS spécifiée.

```
aws ec2 copy-image \
    --source-region us-west-2 \
    --name ami-name \
    --source-image-id ami-066877671789bd71b \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

Sortie :

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

Pour plus d'informations, consultez Copier une AMI dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour inclure vos balises d'AMI définies par l'utilisateur lors de la copie d'une AMI

La `copy-image` commande suivante utilise le `--copy-image-tags` paramètre pour copier les balises AMI définies par l'utilisateur lors de la copie de l'AMI.

```
aws ec2 copy-image \
    --region us-east-1 \
    --name ami-name \
    --source-region us-west-2 \
    --source-image-id ami-066877671789bd71b \
    --description "This is my copied image."
    --copy-image-tags
```

Sortie :

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

Pour plus d'informations, consultez Copier une AMI dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CopyImageà la section Référence des AWS CLI commandes.

`copy-snapshot`

L'exemple de code suivant montre comment utiliser`copy-snapshot`.

AWS CLI

Exemple 1 : pour copier un instantané dans une autre région

L'`copy-snapshot`exemple de commande suivant copie le cliché spécifié de la `us-west-2` région vers la `us-east-1` région et ajoute une brève description.

```
aws ec2 copy-snapshot \
    --region us-east-1 \
    --source-region us-west-2 \
    --source-snapshot-id snap-066877671789bd71b \
    --description 'This is my copied snapshot.'
```

Sortie :

```
{
    "SnapshotId": "snap-066877671789bd71b"
}
```

Exemple 2 : pour copier un instantané non chiffré et chiffrer le nouveau cliché

La `copy-snapshot` commande suivante copie le cliché non chiffré spécifié de la `us-west-2` région vers la région actuelle et chiffre le nouveau cliché à l'aide de la clé KMS spécifiée.

```
aws ec2 copy-snapshot \
    --source-region us-west-2 \
```

```
    --source-snapshot-id snap-066877671789bd71b \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

Sortie :

```
{
    "SnapshotId": "snap-066877671789bd71b"
}
```

Pour plus d'informations, consultez Copier un instantané Amazon EBS dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous CopySnapshotà la section Référence des AWS CLI commandes.

`create-capacity-reservation-fleet`

L'exemple de code suivant montre comment utiliser`create-capacity-reservation-fleet`.

AWS CLI

Pour créer une flotte de réservation de capacité

L'`create-capacity-reservation-fleet`exemple suivant crée une flotte de réservation de capacité pour le type d'instance spécifié dans la demande, jusqu'à la capacité cible totale spécifiée. Le nombre d'instances pour lesquelles la flotte de réserve de capacité réserve de la capacité dépend de la capacité cible totale et des pondérations de type d'instance que vous spécifiez dans la demande. Spécifiez les types d'instances à utiliser et une priorité pour chacun des types d'instances désignés.

```
aws ec2 create-capacity-reservation-fleet \
--total-target-capacity 24 \
--allocation-strategy prioritized \
--instance-match-criteria open \
--tenancy default \
--end-date 2022-12-31T23:59:59.000Z \
--instance-type-specifications file://instanceTypeSpecification.json
```

Contenu de `instanceTypeSpecification.json` :

```
[
    {
        "InstanceType": "m5.xlarge",
        "InstancePlatform": "Linux/UNIX",
        "Weight": 3.0,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 1
    }
]
```

Sortie :

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

Pour plus d'informations sur les flottes de réservation de capacité, consultez la section Flottes de réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

Pour plus d'informations sur le poids du type d'instance et la capacité cible totale, consultez les sections Poids du type d'instance et Capacité cible totale dans le guide de EC2 l'utilisateur Amazon.

Pour plus d'informations sur la désignation de la priorité pour des types d'instances spécifiques, consultez Stratégie d'allocation et Priorité du type d'instance dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateCapacityReservationFleetà la section Référence des AWS CLI commandes.

`create-capacity-reservation`

L'exemple de code suivant montre comment utiliser`create-capacity-reservation`.

AWS CLI

Exemple 1 : pour créer une réservation de capacité

L'`create-capacity-reservation`exemple suivant crée une réservation de capacité dans la zone de `eu-west-1a` disponibilité, dans laquelle vous pouvez lancer trois `t2.medium` instances exécutant un système Linux/Unix d'exploitation. Par défaut, la réservation de capacité est créée avec des critères de correspondance des instances ouvertes et ne prend pas en charge le stockage éphémère, et elle reste active jusqu'à ce que vous l'annuliez manuellement.

```
aws ec2 create-capacity-reservation \
    --availability-zone eu-west-1a \
    --instance-type t2.medium \
    --instance-platform Linux/UNIX \
    --instance-count 3
```

Sortie :

```
{
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
        "EndDateType": "unlimited",
        "AvailabilityZone": "eu-west-1a",
        "InstanceMatchCriteria": "open",
        "EphemeralStorage": false,
        "CreateDate": "2019-08-16T09:27:35.000Z",
        "AvailableInstanceCount": 3,
        "InstancePlatform": "Linux/UNIX",
        "TotalInstanceCount": 3,
        "State": "active",
        "Tenancy": "default",
        "EbsOptimized": false,
        "InstanceType": "t2.medium"
    }
}
```

Exemple 2 : pour créer une réservation de capacité qui se termine automatiquement à une date/heure spécifiée

L'`create-capacity-reservation`exemple suivant crée une réservation de capacité dans la zone de `eu-west-1a` disponibilité, dans laquelle vous pouvez lancer trois `m5.large` instances exécutant un système Linux/Unix d'exploitation. Cette réservation de capacité prend fin automatiquement le 31/08/2019 à 23:59:59.

```
aws ec2 create-capacity-reservation \
```

```
    --availability-zone eu-west-1a \
    --instance-type m5.large \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --end-date-type limited \
    --end-date 2019-08-31T23:59:59Z
```

Sortie :

```
{
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
        "EndDateType": "limited",
        "AvailabilityZone": "eu-west-1a",
        "EndDate": "2019-08-31T23:59:59.000Z",
        "InstanceMatchCriteria": "open",
        "EphemeralStorage": false,
        "CreateDate": "2019-08-16T10:15:53.000Z",
        "AvailableInstanceCount": 3,
        "InstancePlatform": "Linux/UNIX",
        "TotalInstanceCount": 3,
        "State": "active",
        "Tenancy": "default",
        "EbsOptimized": false,
        "InstanceType": "m5.large"
    }
}
```

Exemple 3 : pour créer une réservation de capacité qui accepte uniquement les lancements d'instances ciblés

L'`create-capacity-reservation`exemple suivant crée une réservation de capacité qui accepte uniquement les lancements d'instances ciblés.

```
aws ec2 create-capacity-reservation \
    --availability-zone eu-west-1a \
    --instance-type m5.large \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

Sortie :

```
{
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
        "EndDateType": "unlimited",
        "AvailabilityZone": "eu-west-1a",
        "InstanceMatchCriteria": "targeted",
        "EphemeralStorage": false,
        "CreateDate": "2019-08-16T10:21:57.000Z",
        "AvailableInstanceCount": 3,
        "InstancePlatform": "Linux/UNIX",
        "TotalInstanceCount": 3,
        "State": "active",
        "Tenancy": "default",
        "EbsOptimized": false,
        "InstanceType": "m5.large"
    }
}
```

Pour plus d'informations, consultez la section Créer une réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateCapacityReservationà la section Référence des AWS CLI commandes.

`create-carrier-gateway`

L'exemple de code suivant montre comment utiliser`create-carrier-gateway`.

AWS CLI

Pour créer une passerelle de transporteur

L'`create-carrier-gateway`exemple suivant crée une passerelle de transporteur pour le VPC spécifié.

```
aws ec2 create-carrier-gateway \
    --vpc-id vpc-0c529aEXAMPLE1111
```

Sortie :

```
{
    "CarrierGateway": {
```

```
        "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
        "VpcId": "vpc-0c529aEXAMPLE1111",
        "State": "pending",
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Carrier Gateways dans le guide de l'utilisateur de AWS Wavelength.

- Pour plus de détails sur l'API, reportez-vous CreateCarrierGatewayà la section Référence des AWS CLI commandes.

`create-client-vpn-endpoint`

L'exemple de code suivant montre comment utiliser`create-client-vpn-endpoint`.

AWS CLI

Pour créer un point de terminaison Client VPN

L'`create-client-vpn-endpoint`exemple suivant crée un point de terminaison Client VPN qui utilise l'authentification mutuelle et spécifie une valeur pour le bloc CIDR du client.

```
aws ec2 create-client-vpn-endpoint \
    --client-cidr-block "172.31.0.0/16" \
    --server-certificate-arn arn:aws:acm:ap-south-1:123456789012:certificate/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --authentication-options Type=certificate-
authentication,MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE} \
    --connection-log-options Enabled=false
```

Sortie :

```
{
    "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
    "Status": {
        "Code": "pending-associate"
    },
    "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com"
```

```
}
```

Pour plus d'informations, consultez la section Points de terminaison du VPN client dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous CreateClientVpnEndpointà la section Référence des AWS CLI commandes.

`create-client-vpn-route`

L'exemple de code suivant montre comment utiliser`create-client-vpn-route`.

AWS CLI

Pour créer un itinéraire pour un point de terminaison VPN client

L'`create-client-vpn-route`exemple suivant ajoute une route vers Internet (`0.0.0.0/0`) pour le sous-réseau spécifié du point de terminaison VPN du Client.

```
aws ec2 create-client-vpn-route \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --destination-cidr-block 0.0.0.0/0  \
    --target-vpc-subnet-id subnet-0123456789abcabca
```

Sortie :

```
{
    "Status": {
        "Code": "creating"
    }
}
```

Pour plus d'informations, consultez Routes dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous CreateClientVpnRouteà la section Référence des AWS CLI commandes.

`create-coip-cidr`

L'exemple de code suivant montre comment utiliser`create-coip-cidr`.

AWS CLI

Pour créer une plage d'adresses IP (CoIP) appartenant au client

L'`create-coip-cidr`exemple suivant crée la plage d'adresses CoIP spécifiée dans le pool CoIP spécifié.

```
aws ec2 create-coip-cidr \
    --cidr 15.0.0.0/24 \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Sortie :

```
{
    "CoipCidr": {
        "Cidr": "15.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

Pour plus d'informations, voir Adresses IP appartenant au client dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous CreateCoipCidrà la section Référence des AWS CLI commandes.

`create-coip-pool`

L'exemple de code suivant montre comment utiliser`create-coip-pool`.

AWS CLI

Pour créer un pool d'adresses IP (CoIP) appartenant au client

L'`create-coip-pool`exemple suivant crée un pool CoIP pour les adresses CoIP dans la table de routage de passerelle locale spécifiée.

```
aws ec2 create-coip-pool \
    --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Sortie :

```
{
    "CoipPool": {
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
    }
}
```

Pour plus d'informations, voir Adresses IP appartenant au client dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous CreateCoipPoolà la section Référence des AWS CLI commandes.

`create-customer-gateway`

L'exemple de code suivant montre comment utiliser`create-customer-gateway`.

AWS CLI

Pour créer une passerelle client

Cet exemple crée une passerelle client avec l'adresse IP spécifiée pour son interface externe.

Commande :

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

Sortie :

```
{
    "CustomerGateway": {
        "CustomerGatewayId": "cgw-0e11f167",
        "IpAddress": "12.1.2.3",
        "State": "available",
        "Type": "ipsec.1",
        "BgpAsn": "65534"
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateCustomerGatewayà la section Référence des AWS CLI commandes.

`create-default-subnet`

L'exemple de code suivant montre comment utiliser`create-default-subnet`.

AWS CLI

Pour créer un sous-réseau par défaut

Cet exemple crée un sous-réseau par défaut dans la zone `us-east-2a` de disponibilité.

Commande :

```
 aws ec2 create-default-subnet --availability-zone us-east-2a

{
   "Subnet": {
       "AvailabilityZone": "us-east-2a",
       "Tags": [],
       "AvailableIpAddressCount": 4091,
       "DefaultForAz": true,
       "Ipv6CidrBlockAssociationSet": [],
       "VpcId": "vpc-1a2b3c4d",
       "State": "available",
       "MapPublicIpOnLaunch": true,
       "SubnetId": "subnet-1122aabb",
       "CidrBlock": "172.31.32.0/20",
       "AssignIpv6AddressOnCreation": false
   }
 }
```

- Pour plus de détails sur l'API, reportez-vous CreateDefaultSubnetà la section Référence des AWS CLI commandes.

`create-default-vpc`

L'exemple de code suivant montre comment utiliser`create-default-vpc`.

AWS CLI

Pour créer un VPC par défaut

Cet exemple crée un VPC par défaut.

Commande :

```
aws ec2 create-default-vpc
```

Sortie :

```
{
   "Vpc": {
       "VpcId": "vpc-8eaae5ea",
       "InstanceTenancy": "default",
       "Tags": [],
       "Ipv6CidrBlockAssociationSet": [],
       "State": "pending",
       "DhcpOptionsId": "dopt-af0c32c6",
       "CidrBlock": "172.31.0.0/16",
       "IsDefault": true
   }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDefaultVpcà la section Référence des AWS CLI commandes.

`create-dhcp-options`

L'exemple de code suivant montre comment utiliser`create-dhcp-options`.

AWS CLI

Pour créer un ensemble d'options DHCP

L'`create-dhcp-options`exemple suivant crée un ensemble d'options DHCP qui spécifie le nom de domaine, les serveurs de noms de domaine et le type de nœud NetBIOS.

```
aws ec2 create-dhcp-options \
    --dhcp-configuration \
```

```
        "Key=domain-name-servers,Values=10.2.5.1,10.2.5.2" \
        "Key=domain-name,Values=example.com" \
        "Key=netbios-node-type,Values=2"
```

Sortie :

```
{
    "DhcpOptions": {
        "DhcpConfigurations": [
            {
                "Key": "domain-name",
                "Values": [
                    {
                        "Value": "example.com"
                    }
                ]
            },
            {
                "Key": "domain-name-servers",
                "Values": [
                    {
                        "Value": "10.2.5.1"
                    },
                    {
                        "Value": "10.2.5.2"
                    }
                ]
            },
            {
                "Key": "netbios-node-type",
                "Values": [
                    {
                        "Value": "2"
                    }
                ]
            }
        ],
        "DhcpOptionsId": "dopt-06d52773eff4c55f3"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDhcpOptionsà la section Référence des AWS CLI commandes.

`create-egress-only-internet-gateway`

L'exemple de code suivant montre comment utiliser`create-egress-only-internet-gateway`.

AWS CLI

Pour créer une passerelle Internet de sortie uniquement

Cet exemple crée une passerelle Internet de sortie uniquement pour le VPC spécifié.

Commande :

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

Sortie :

```
{
  "EgressOnlyInternetGateway": {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [
          {
              "State": "attached",
              "VpcId": "vpc-0c62a468"
          }
      ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateEgressOnlyInternetGatewayà la section Référence des AWS CLI commandes.

`create-fleet`

L'exemple de code suivant montre comment utiliser`create-fleet`.

AWS CLI

Exemple 1 : pour créer une EC2 flotte qui lance les instances Spot comme modèle d'achat par défaut

L'`create-fleet`exemple suivant crée une EC2 flotte en utilisant les paramètres minimaux requis pour lancer une flotte : un modèle de lancement, une capacité cible et un modèle d'achat par

défaut. Le modèle de lancement est identifié par son ID de modèle de lancement et son numéro de version. La capacité cible de la flotte est de 2 instances, et le modèle d'achat par défaut est`spot`, ce qui permet au parc de lancer 2 instances ponctuelles.

Lorsque vous créez une EC2 flotte, utilisez un fichier JSON pour spécifier les informations relatives aux instances à lancer.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

Contenu du fichier file_name.json :

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Sortie :

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Exemple 2 : pour créer une EC2 flotte qui lance des instances à la demande comme modèle d'achat par défaut

L'`create-fleet`exemple suivant crée une EC2 flotte en utilisant les paramètres minimaux requis pour lancer une flotte : un modèle de lancement, une capacité cible et un modèle d'achat par défaut. Le modèle de lancement est identifié par son ID de modèle de lancement et son numéro de version. La capacité cible du parc est de 2 instances, et le modèle d'achat par défaut est`on-demand`, ce qui permet au parc de lancer 2 instances à la demande.

Lorsque vous créez une EC2 flotte, utilisez un fichier JSON pour spécifier les informations relatives aux instances à lancer.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

Contenu du fichier file_name.json :

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
    }
}
```

Sortie :

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Exemple 3 : pour créer une EC2 flotte qui lance des instances à la demande en tant que capacité principale

L'`create-fleet`exemple suivant crée un EC2 parc qui indique la capacité cible totale de 2 instances pour le parc et une capacité cible d'une instance à la demande. Le modèle d'achat par défaut est`spot`. La flotte lance 1 instance à la demande comme indiqué, mais doit en lancer une autre pour atteindre la capacité cible totale. Le modèle d'achat correspondant à la différence est calculé comme suit `TotalTargetCapacity` : - `OnDemandTargetCapacity` =`DefaultTargetCapacityType`, ce qui permet à la flotte de lancer une instance Spot.

Lorsque vous créez une EC2 flotte, utilisez un fichier JSON pour spécifier les informations relatives aux instances à lancer.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

Contenu du fichier file_name.json :

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity":1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Sortie :

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Exemple 4 : créer une EC2 flotte qui lance des instances ponctuelles en utilisant la stratégie d'allocation du prix le plus bas

Si la stratégie d'allocation pour les Instances Spot n'est pas spécifiée, la stratégie d'allocation par défaut, à savoir `lowest-price`, est utilisée. L'`create-fleet`exemple suivant crée une EC2 flotte à l'aide de la stratégie `lowest-price` d'allocation. Les trois spécifications de lancement, qui remplacent le modèle de lancement, ont des types d'instance différents mais la même capacité pondérée et le même sous-réseau. La capacité cible totale est de 2 instances et le modèle d'achat par défaut est`spot`. La EC2 flotte lance 2 instances ponctuelles en utilisant le type d'instance indiqué dans la spécification de lancement au prix le plus bas.

Lorsque vous créez une EC2 flotte, utilisez un fichier JSON pour spécifier les informations relatives aux instances à lancer.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.jsonContents of file_name.json::

{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        },
        "Overrides": [
            {
                "InstanceType": "c4.large",
                "WeightedCapacity": 1,
                "SubnetId": "subnet-a4f6c5d3"
            },
            {
                "InstanceType": "c3.large",
                "WeightedCapacity": 1,
                "SubnetId": "subnet-a4f6c5d3"
            },
            {
                "InstanceType": "c5.large",
                "WeightedCapacity": 1,
                "SubnetId": "subnet-a4f6c5d3"
            }
        ]
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"

  }
}
```

Sortie :

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFleetà la section Référence des AWS CLI commandes.

`create-flow-logs`

L'exemple de code suivant montre comment utiliser`create-flow-logs`.

AWS CLI

Exemple 1 : pour créer un journal de flux

L'`create-flow-logs`exemple suivant crée un journal de flux qui capture tout le trafic rejeté pour l'interface réseau spécifiée. Les journaux de flux sont transmis à un groupe de CloudWatch journaux dans Logs à l'aide des autorisations associées au rôle IAM spécifié.

```
aws ec2 create-flow-logs \
    --resource-type NetworkInterface \
    --resource-ids eni-11223344556677889 \
    --traffic-type REJECT \
    --log-group-name my-flow-logs \
    --deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
```

Sortie :

```
{
    "ClientToken": "so0eNA2uSHUNlHI0S2cJ305GuIX1CezaRdGtexample",
    "FlowLogIds": [
        "fl-12345678901234567"
    ],
    "Unsuccessful": []
}
```

Pour plus d'informations, consultez la rubrique Journaux de flux VPC dans le Guide de l'utilisateur Amazon VPC.

Exemple 2 : pour créer un journal de flux avec un format personnalisé

L'`create-flow-logs`exemple suivant crée un journal de flux qui capture tout le trafic pour le VPC spécifié et transmet les journaux de flux à un compartiment Amazon S3. Le paramètre `--log-format` spécifie un format personnalisé pour les enregistrements de journal de flux. Pour

exécuter cette commande sous Windows, remplacez les guillemets simples (') par des guillemets doubles («).

```
aws ec2 create-flow-logs \
    --resource-type VPC \
    --resource-ids vpc-00112233344556677 \
    --traffic-type ALL \
    --log-destination-type s3 \
    --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
    --log-format '${version} ${vpc-id} ${subnet-id} ${instance-id} ${srcaddr}
 ${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-srcaddr}
 ${pkt-dstaddr}'
```

Pour plus d'informations, consultez la rubrique Journaux de flux VPC dans le Guide de l'utilisateur Amazon VPC.

Exemple 3 : pour créer un journal de flux avec un intervalle d'agrégation maximal d'une minute

L'`create-flow-logs`exemple suivant crée un journal de flux qui capture tout le trafic pour le VPC spécifié et transmet les journaux de flux à un compartiment Amazon S3. Le `--max-aggregation-interval` paramètre spécifie un intervalle d'agrégation maximal de 60 secondes (1 minute).

```
aws ec2 create-flow-logs \
    --resource-type VPC \
    --resource-ids vpc-00112233344556677 \
    --traffic-type ALL \
    --log-destination-type s3 \
    --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
    --max-aggregation-interval 60
```

Pour plus d'informations, consultez la rubrique Journaux de flux VPC dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous CreateFlowLogsà la section Référence des AWS CLI commandes.

`create-fpga-image`

L'exemple de code suivant montre comment utiliser`create-fpga-image`.

AWS CLI

Pour créer une image Amazon FPGA

Cet exemple crée un fichier AFI à partir de l'archive tar spécifiée dans le compartiment spécifié.

Commande :

```
aws ec2 create-fpga-image --name my-afi --description test-afi --input-storage-location Bucket=my-fpga-bucket,Key=dcp/17_12_22-103226.Developer_CL.tar --logs-storage-location Bucket=my-fpga-bucket,Key=logs
```

Sortie :

```
{
  "FpgaImageId": "afi-0d123e123bfc85abc",
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFpgaImageà la section Référence des AWS CLI commandes.

`create-image`

L'exemple de code suivant montre comment utiliser`create-image`.

AWS CLI

Exemple 1 : pour créer une AMI à partir d'une instance basée sur Amazon EBS

L'`create-image`exemple suivant crée une AMI à partir de l'instance spécifiée.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --description "An AMI for my server"
```

Sortie :

```
{
```

```
    "ImageId": "ami-abcdef01234567890"
}
```

Pour plus d'informations sur la spécification d'un mappage de périphérique en mode bloc pour votre AMI, consultez la section Spécification d'un mappage de périphérique en mode bloc pour une AMI dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour créer une AMI à partir d'une instance basée sur Amazon EBS sans redémarrer

L'`create-image`exemple suivant crée une AMI et définit le paramètre --no-reboot afin que l'instance ne soit pas redémarrée avant la création de l'image.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --no-reboot
```

Sortie :

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

Pour plus d'informations sur la spécification d'un mappage de périphérique en mode bloc pour votre AMI, consultez la section Spécification d'un mappage de périphérique en mode bloc pour une AMI dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour étiqueter une AMI et des instantanés lors de leur création

L'`create-image`exemple suivant crée une AMI et étiquette l'AMI et les instantanés avec la même balise. `cost-center=cc123`

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --tag-specifications "ResourceType=image,Tags=[{Key=cost-
center,Value=cc123}]" "ResourceType=snapshot,Tags=[{Key=cost-center,Value=cc123}]"
```

Sortie :

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

Pour plus d'informations sur le balisage de vos ressources lors de leur création, consultez la section Ajouter des balises lors de la création de ressources dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateImageà la section Référence des AWS CLI commandes.

`create-instance-connect-endpoint`

L'exemple de code suivant montre comment utiliser`create-instance-connect-endpoint`.

AWS CLI

Pour créer un point de terminaison EC2 Instance Connect

L'`create-instance-connect-endpoint`exemple suivant crée un point de terminaison EC2 Instance Connect dans le sous-réseau spécifié.

```
aws ec2 create-instance-connect-endpoint \
    --region us-east-1 \
    --subnet-id subnet-0123456789example
```

Sortie :

```
{
    "VpcId": "vpc-0123abcd",
    "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
    "AvailabilityZone": "us-east-1a",
    "NetworkInterfaceIds": [
        "eni-0123abcd"
    ],
    "PreserveClientIp": true,
    "Tags": [],
    "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
    "StateMessage": "",
```

```
    "State": "create-complete",
    "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-endpoint.us-
east-1.amazonaws.com",
    "SubnetId": "subnet-0123abcd",
    "OwnerId": "111111111111",
    "SecurityGroupIds": [
        "sg-0123abcd"
    ],
    "InstanceConnectEndpointId": "eice-0123456789example",
    "CreatedAt": "2023-04-07T15:43:53.000Z"
}
```

Pour plus d'informations, consultez Create an EC2 Instance Connect Endpoint dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateInstanceConnectEndpointà la section Référence des AWS CLI commandes.

`create-instance-event-window`

L'exemple de code suivant montre comment utiliser`create-instance-event-window`.

AWS CLI

Exemple 1 : pour créer une fenêtre d'événements avec une plage de temps

L'`create-instance-event-window`exemple suivant crée une fenêtre d'événements avec une plage de temps. Vous ne pouvez pas également spécifier le paramètre `cron-expression`.

```
aws ec2 create-instance-event-window \
    --region us-east-1 \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
```

```
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour créer une fenêtre d'événements avec une expression cron

L'`create-instance-event-window`exemple suivant crée une fenêtre d'événements avec une expression cron. Vous ne pouvez pas également spécifier le paramètre `time-range`.

```
aws ec2 create-instance-event-window \
    --region us-east-1 \
    --cron-expression "* 21-23 * * 2,3" \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
```

```
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateInstanceEventWindowà la section Référence des AWS CLI commandes.

`create-instance-export-task`

L'exemple de code suivant montre comment utiliser`create-instance-export-task`.

AWS CLI

Pour exporter une instance

Cet exemple de commande crée une tâche pour exporter l'instance i-1234567890abcdef0 vers le compartiment myexportbucket d'Amazon S3.

Commande :

```
aws ec2 create-instance-export-task --description "RHEL5 instance" --
instance-id i-1234567890abcdef0 --target-environment vmware --export-to-s3-
task DiskImageFormat=vmdk,ContainerFormat=ova,S3Bucket=myexportbucket,S3Prefix=RHEL5
```

Sortie :

```
{
    "ExportTask": {
        "State": "active",
        "InstanceExportDetails": {
            "InstanceId": "i-1234567890abcdef0",
            "TargetEnvironment": "vmware"
        },
```

```
        "ExportToS3Task": {
            "S3Bucket": "myexportbucket",
            "S3Key": "RHEL5export-i-fh8sjjsq.ova",
            "DiskImageFormat": "vmdk",
            "ContainerFormat": "ova"
        },
        "Description": "RHEL5 instance",
        "ExportTaskId": "export-i-fh8sjjsq"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateInstanceExportTaskà la section Référence des AWS CLI commandes.

`create-internet-gateway`

L'exemple de code suivant montre comment utiliser`create-internet-gateway`.

AWS CLI

Pour créer une passerelle Internet

L'`create-internet-gateway`exemple suivant crée une passerelle Internet avec la baliseName=my-igw.

```
aws ec2 create-internet-gateway \
    --tag-specifications ResourceType=internet-gateway,Tags=[{Key=Name,Value=my-igw}]
```

Sortie :

```
{
    "InternetGateway": {
        "Attachments": [],
        "InternetGatewayId": "igw-0d0fb496b3994d755",
        "OwnerId": "123456789012",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-igw"
            }
```

```
        ]
    }
}
```

Pour plus d'informations, consultez Passerelles Internet dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous CreateInternetGatewayà la section Référence des AWS CLI commandes.

`create-ipam-pool`

L'exemple de code suivant montre comment utiliser`create-ipam-pool`.

AWS CLI

Pour créer un pool IPAM

L'`create-ipam-pool`exemple suivant crée un pool IPAM.

(Linux) :

```
aws ec2 create-ipam-pool \
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
    --address-family ipv4 \
    --auto-import \
    --allocation-min-netmask-length 16 \
    --allocation-max-netmask-length 26 \
    --allocation-default-netmask-length 24 \
    --allocation-resource-tags "Key=Environment,Value=Preprod" \
    --tag-specifications 'ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod
 pool"}]'
```

(Fenêtres) :

```
aws ec2 create-ipam-pool ^
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
    --address-family ipv4 ^
    --auto-import ^
    --allocation-min-netmask-length 16 ^
    --allocation-max-netmask-length 26 ^
```

```
    --allocation-default-netmask-length 24 ^
    --allocation-resource-tags "Key=Environment,Value=Preprod" ^
    --tag-specifications ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod
 pool"}]
```

Sortie :

```
{
    "IpamPool": {
        "OwnerId": "123456789012",
        "IpamPoolId": "ipam-pool-0533048da7d823723",
        "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-0533048da7d823723",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
        "IpamScopeType": "private",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "Locale": "None",
        "PoolDepth": 1,
        "State": "create-in-progress",
        "AutoImport": true,
        "AddressFamily": "ipv4",
        "AllocationMinNetmaskLength": 16,
        "AllocationMaxNetmaskLength": 26,
        "AllocationDefaultNetmaskLength": 24,
        "AllocationResourceTags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            }
        ],
        "Tags": [
            {
                "Key": "Name",
                "Value": "Preprod pool"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Planifier le provisionnement des adresses IP dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous CreateIpamPoolà la section Référence des AWS CLI commandes.

`create-ipam-resource-discovery`

L'exemple de code suivant montre comment utiliser`create-ipam-resource-discovery`.

AWS CLI

Pour créer une découverte de ressources

Dans cet exemple, vous êtes un administrateur IPAM délégué qui souhaite créer et partager une découverte de ressources avec l'administrateur IPAM d'une autre AWS organisation afin que celui-ci puisse gérer et surveiller les adresses IP des ressources de votre organisation.

Important

Cet exemple inclut à la fois les `--operating-regions` options `--region` et car, bien qu'elles soient facultatives, elles doivent être configurées d'une manière particulière pour intégrer correctement une découverte de ressources à un IPAM. * `--operating-regions` doit correspondre aux régions dans lesquelles vous souhaitez que l'IPAM découvre des ressources. S'il existe des régions dans lesquelles vous ne souhaitez pas que l'IPAM gère les adresses IP (par exemple pour des raisons de conformité), ne les incluez pas. * `--region` doit correspondre à la région d'origine de l'IPAM à laquelle vous souhaitez l'associer. Vous devez créer la découverte des ressources dans la même région que celle dans laquelle l'IPAM a été créé. Par exemple, si l'IPAM auquel vous vous associez a été créé dans us-east-1, `--region us-east-1` incluez-le dans la demande. Les options `--operating-regions` et par défaut correspondent à la région dans laquelle vous exécutez `--region` la commande si vous ne les spécifiez pas.

Dans cet exemple, les régions opérationnelles de l'IPAM auxquelles nous procédons à l'intégration incluent `us-west-1`us-west-2, et`ap-south-1`. Lorsque nous créons la découverte des ressources, nous voulons que l'IPAM découvre les adresses IP des ressources dans `us-west-1` et `us-west-2` non`ap-south-1`. Nous n'incluons donc que `--operating-regions RegionName='us-west-1' RegionName='us-west-2'` dans la demande.

L'`create-ipam-resource-discovery`exemple suivant crée une découverte de ressources IPAM.

```
aws ec2 create-ipam-resource-discovery \
```

```
    --description 'Example-resource-discovery' \
    --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-
center,Value=cc123}]' \
    --operating-regions RegionName='us-west-1' RegionName='us-west-2' \
    --region us-east-1
```

Sortie :

```
{
    "IpamResourceDiscovery":{
        "OwnerId": "149977607591",
        "IpamResourceDiscoveryId": "ipam-res-disco-0257046d8aa78b8bc",
        "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0257046d8aa78b8bc",
        "IpamResourceDiscoveryRegion": "us-east-1",
        "Description": "'Example-resource-discovery'",
        "OperatingRegions":[
            {"RegionName": "us-west-1"},
            {"RegionName": "us-west-2"},
            {"RegionName": "us-east-1"}
        ],
        "IsDefault": false,
        "State": "create-in-progress",
        "Tags": [
            {
                "Key": "cost-center",
                "Value": "cc123"
            }
        ]
}
```

Une fois que vous avez créé une découverte de ressources, vous souhaiterez peut-être la partager avec un autre administrateur délégué IPAM, ce que vous pourrez faire avec create-resource-share. Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous CreateIpamResourceDiscoveryà la section Référence des AWS CLI commandes.

`create-ipam-scope`

L'exemple de code suivant montre comment utiliser`create-ipam-scope`.

AWS CLI

Pour créer un scope IPAM

L'`create-ipam-scope`exemple suivant crée une étendue IPAM.

(Linux) :

```
aws ec2 create-ipam-scope \
    --ipam-id ipam-08440e7a3acde3908 \
    --description "Example description" \
    --tag-specifications 'ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example
 name value"}]'
```

(Fenêtres) :

```
aws ec2 create-ipam-scope ^
    --ipam-id ipam-08440e7a3acde3908 ^
    --description "Example description" ^
    --tag-specifications ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example name
 value"}]
```

Sortie :

```
{
    "IpamScope": {
        "OwnerId": "123456789012",
        "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-01c1ebab2b63bd7e4",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "IpamScopeType": "private",
        "IsDefault": false,
        "Description": "Example description",
        "PoolCount": 0,
        "State": "create-in-progress",
        "Tags": [
            {
                "Key": "Name",
                "Value": "Example name value"
            }
```

```
        ]
    }
}
```

Pour plus d'informations, consultez la section Créer des étendues supplémentaires dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous CreateIpamScopeà la section Référence des AWS CLI commandes.

`create-ipam`

L'exemple de code suivant montre comment utiliser`create-ipam`.

AWS CLI

Pour créer un IPAM

L'`create-ipam`exemple suivant crée un IPAM.

(Linux) :

```
aws ec2 create-ipam \
    --description "Example description" \
    --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" \
    --tag-specifications 'ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]'
```

(Fenêtres) :

```
aws ec2 create-ipam ^
    --description "Example description" ^
    --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" ^
    --tag-specifications ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]
```

Sortie :

```
{
    "Ipam": {
        "OwnerId": "123456789012",
        "IpamId": "ipam-036486dfa6af58ee0",
```

```
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
        "IpamRegion": "us-east-1",
        "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
        "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
        "ScopeCount": 2,
        "OperatingRegions": [
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-west-1"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "State": "create-in-progress",
        "Tags": [
            {
                "Key": "Name",
                "Value": "ExampleIPAM"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Créer un IPAM dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous CreateIpamà la section Référence des AWS CLI commandes.

`create-key-pair`

L'exemple de code suivant montre comment utiliser`create-key-pair`.

AWS CLI

Création d'une paire de clés

Cet exemple crée une paire de clés nommée `MyKeyPair`.

Commande :

```
aws ec2 create-key-pair --key-name MyKeyPair
```

La sortie est une version ASCII de la clé privée et de l'empreinte de la clé. Vous devez enregistrer la clé dans un fichier.

Pour plus d'informations, veuillez consulter la rubrique Using Key Pairs dans le Guide de l'utilisateur AWS Command Line Interface.

- Pour plus de détails sur l'API, reportez-vous CreateKeyPairà la section Référence des AWS CLI commandes.

`create-launch-template-version`

L'exemple de code suivant montre comment utiliser`create-launch-template-version`.

AWS CLI

Pour créer une version du modèle de lancement

Cet exemple crée une nouvelle version du modèle de lancement basée sur la version 1 du modèle de lancement et spécifie un ID d'AMI différent.

Commande :

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
 --version-description WebVersion2 --source-version 1 --launch-template-data
 '{"ImageId":"ami-c998b6b2"}'
```

Sortie :

```
{
  "LaunchTemplateVersion": {
      "VersionDescription": "WebVersion2",
      "LaunchTemplateId": "lt-0abcd290751193123",
      "LaunchTemplateName": "WebServers",
      "VersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "LaunchTemplateData": {
          "ImageId": "ami-c998b6b2",
          "InstanceType": "t2.micro",
```

```
            "NetworkInterfaces": [
                {
                    "Ipv6Addresses": [
                        {
                            "Ipv6Address": "2001:db8:1234:1a00::123"
                        }
                    ],
                    "DeviceIndex": 0,
                    "SubnetId": "subnet-7b16de0c",
                    "AssociatePublicIpAddress": true
                }
            ]
        },
        "DefaultVersion": false,
        "CreateTime": "2017-12-01T13:35:46.000Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLaunchTemplateVersionà la section Référence des AWS CLI commandes.

`create-launch-template`

L'exemple de code suivant montre comment utiliser`create-launch-template`.

AWS CLI

Exemple 1 : création d'un modèle de lancement

L'`create-launch-template`exemple suivant crée un modèle de lancement qui indique le sous-réseau dans lequel lancer l'instance, attribue une adresse IP publique et une IPv6 adresse à l'instance, et crée une balise pour l'instance.

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --version-description WebVersion1 \
    --launch-template-data '{"NetworkInterfaces":
[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1,"SubnetId":"subnet-7b
[{"ResourceType":"instance","Tags":[{"Key":"purpose","Value":"webserver"}]}]}'
```

Sortie :

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-01-27T09:13:24.000Z"
    }
}
```

Pour de plus amples informations, veuillez consulter la rubrique Lancer une instance à partir d'un modèle de lancement dans le Guide de l'utilisateur Amazon Elastic Compute Cloud. Pour plus d'informations sur l'utilisation de guillemets avec les paramètres au format JSON, veuillez consulter la rubrique Quoting Strings dans le Guide de l'utilisateur AWS Command Line Interface.

Exemple 2 : pour créer un modèle de lancement pour Amazon EC2 Auto Scaling

L'exemple `create-launch-template` suivant crée un modèle de lancement avec plusieurs balises et un mappage de périphérique de stockage en mode bloc pour spécifier un volume EBS supplémentaire lors du lancement d'une instance. Spécifiez une valeur pour `Groups` correspondant à des groupes de sécurité pour le VPC dans lequel votre groupe Auto Scaling lancera des instances. Spécifiez le VPC et les sous-réseaux en tant que propriétés du groupe Auto Scaling.

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForAutoScaling \
    --version-description AutoScalingVersion1 \
    --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
["sg-7c227019,sg-903004f8"],"DeleteOnTermination":true}],"ImageId":"ami-
b42209de","InstanceType":"m4.large","TagSpecifications":
[{"ResourceType":"instance","Tags":[{"Key":"environment","Value":"production"},
{"Key":"purpose","Value":"webserver"}]},{"ResourceType":"volume","Tags":
[{"Key":"environment","Value":"production"},{"Key":"cost-
center","Value":"cc123"}]}],"BlockDeviceMappings":[{"DeviceName":"/dev/sda1","Ebs":
{"VolumeSize":100}}]}' --region us-east-1
```

Sortie :

```
{
```

```
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0123c79c33a54e0abc",
        "LaunchTemplateName": "TemplateForAutoScaling",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-04-30T18:16:06.000Z"
    }
}
```

Pour plus d'informations, consultez la section Création d'un modèle de lancement pour un groupe Auto Scaling dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling. Pour plus d'informations sur l'utilisation de guillemets avec les paramètres au format JSON, veuillez consulter la rubrique Quoting Strings dans le Guide de l'utilisateur AWS Command Line Interface.

Exemple 3 : création d'un modèle de lancement qui spécifie le chiffrement des volumes EBS

L'exemple `create-launch-template` suivant crée un modèle de lancement qui inclut des volumes EBS chiffrés créés à partir d'un instantané non chiffré. Il étiquette également les volumes lors de leur création. Si le chiffrement est désactivé par défaut, vous devez spécifier l'option `"Encrypted"` comme indiqué dans l'exemple suivant. Si vous utilisez l'option `"KmsKeyId"` pour spécifier une clé CMK gérée par le client, vous devez également spécifier l'option `"Encrypted"`, même si le chiffrement est activé par défaut.

```
aws ec2 create-launch-template \
  --launch-template-name TemplateForEncryption \
  --launch-template-data file://config.json
```

Contenu de `config.json` :

```
{
    "BlockDeviceMappings":[
        {
            "DeviceName":"/dev/sda1",
            "Ebs":{
                "VolumeType":"gp2",
                "DeleteOnTermination":true,
                "SnapshotId":"snap-066877671789bd71b",
                "Encrypted":true,
                "KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/abcd1234-
a123-456a-a12b-a123b4cd56ef"
```

```
            }
        }
    ],
    "ImageId":"ami-00068cd7555f543d5",
    "InstanceType":"c5.large",
    "TagSpecifications":[
        {
            "ResourceType":"volume",
            "Tags":[
                {
                    "Key":"encrypted",
                    "Value":"yes"
                }
            ]
        }
    ]
}
```

Sortie :

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
        "LaunchTemplateName": "TemplateForEncryption",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2020-01-07T19:08:36.000Z"
    }
}
```

Pour de plus amples informations, veuillez consulter la rubrique les rubriques Restauration d'un volume Amazon EBS et Chiffrement par défaut dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous CreateLaunchTemplateà la section Référence des AWS CLI commandes.

`create-local-gateway-route-table-virtual-interface-group-association`

L'exemple de code suivant montre comment utiliser`create-local-gateway-route-table-virtual-interface-group-association`.

AWS CLI

Pour associer une table de routage de passerelle locale à un groupe d'interfaces virtuelles (VIFs)

L'`create-local-gateway-route-table-virtual-interface-group-association`exemple suivant crée une association entre la table de routage de passerelle locale spécifiée et le groupe VIF.

```
aws ec2 create-local-gateway-route-table-virtual-interface-group-association \
    --local-gateway-route-table-id lgw-rtb-exampleidabcd1234 \
    --local-gateway-virtual-interface-group-id lgw-vif-grp-exampleid0123abcd
```

Sortie :

```
{
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {
        "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-
assoc-exampleid12345678",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",
        "LocalGatewayId": "lgw-exampleid11223344",
        "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-exampleidabcd1234",
        "OwnerId": "111122223333",
        "State": "pending",
        "Tags": []
    }
}
```

Pour plus d'informations, consultez les associations de groupes VIF dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociationà la section Référence des AWS CLI commandes.

`create-local-gateway-route-table-vpc-association`

L'exemple de code suivant montre comment utiliser`create-local-gateway-route-table-vpc-association`.

AWS CLI

Pour associer un VPC à une table de routage

L'`create-local-gateway-route-table-vpc-association`exemple suivant associe le VPC spécifié à la table de routage de passerelle locale spécifiée.

```
aws ec2 create-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --vpc-id vpc-07ef66ac71EXAMPLE
```

Sortie :

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-07ef66ac71EXAMPLE",
        "State": "associated"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLocalGatewayRouteTableVpcAssociationà la section Référence des AWS CLI commandes.

`create-local-gateway-route-table`

L'exemple de code suivant montre comment utiliser`create-local-gateway-route-table`.

AWS CLI

Pour créer une table de routage de passerelle locale

L'`create-local-gateway-route-table`exemple suivant crée une table de routage de passerelle locale avec le mode de routage VPC direct.

```
aws ec2 create-local-gateway-route-table \
    --local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \
    --mode direct-vpc-routing
```

Sortie :

```
{
    "LocalGatewayRouteTable": {
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
        "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
        "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-021345abcdef67890",
        "OwnerId": "111122223333",
        "State": "pending",
        "Tags": [],
        "Mode": "direct-vpc-routing"
    }
}
```

Pour en savoir plus, consultez Tables de routage de passerelle locale dans le Guide de l'utilisateur AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous CreateLocalGatewayRouteTableà la section Référence des AWS CLI commandes.

`create-local-gateway-route`

L'exemple de code suivant montre comment utiliser`create-local-gateway-route`.

AWS CLI

Pour créer un itinéraire statique pour une table de routage de passerelle locale

L'`create-local-gateway-route`exemple suivant crée l'itinéraire spécifié dans la table de routage de passerelle locale spécifiée.

```
aws ec2 create-local-gateway-route \
    --destination-cidr-block 0.0.0.0/0 \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

Sortie :

```
{
```

```
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLocalGatewayRouteà la section Référence des AWS CLI commandes.

`create-managed-prefix-list`

L'exemple de code suivant montre comment utiliser`create-managed-prefix-list`.

AWS CLI

Pour créer une liste de préfixes

L'`create-managed-prefix-list`exemple suivant crée une liste de IPv4 préfixes avec un maximum de 10 entrées, et crée 2 entrées dans la liste de préfixes.

```
aws ec2 create-managed-prefix-list \
    --address-family IPv4 \
    --max-entries 10 \
    --entries Cidr=10.0.0.0/16,Description=vpc-a Cidr=10.2.0.0/16,Description=vpc-b
 \
    --prefix-list-name vpc-cidrs
```

Sortie :

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "create-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 1,
```

```
        "Tags": [],
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous CreateManagedPrefixListà la section Référence des AWS CLI commandes.

`create-nat-gateway`

L'exemple de code suivant montre comment utiliser`create-nat-gateway`.

AWS CLI

Exemple 1 : pour créer une passerelle NAT publique

L'`create-nat-gateway`exemple suivant crée une passerelle NAT publique dans le sous-réseau spécifié et associe l'adresse IP élastique à l'ID d'allocation spécifié. Lorsque vous créez une passerelle NAT publique, vous devez associer une adresse IP élastique.

```
aws ec2 create-nat-gateway \
    --subnet-id subnet-0250c25a1fEXAMPLE \
    --allocation-id eipalloc-09ad461b0dEXAMPLE
```

Sortie :

```
{
    "NatGateway": {
        "CreateTime": "2021-12-01T22:22:38.000Z",
        "NatGatewayAddresses": [
            {
                "AllocationId": "eipalloc-09ad461b0dEXAMPLE"
            }
        ],
        "NatGatewayId": "nat-0c61bf8a12EXAMPLE",
        "State": "pending",
        "SubnetId": "subnet-0250c25a1fEXAMPLE",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
```

```
        "ConnectivityType": "public"
    }
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

Exemple 2 : pour créer une passerelle NAT privée

L'`create-nat-gateway`exemple suivant crée une passerelle NAT privée dans le sous-réseau spécifié. Aucune adresse IP élastique n'est associée à une passerelle NAT privée.

```
aws ec2 create-nat-gateway \
    --subnet-id subnet-0250c25a1fEXAMPLE \
    --connectivity-type private
```

Sortie :

```
{
    "NatGateway": {
        "CreateTime": "2021-12-01T22:26:00.000Z",
        "NatGatewayAddresses": [
            {}
        ],
        "NatGatewayId": "nat-011b568379EXAMPLE",
        "State": "pending",
        "SubnetId": "subnet-0250c25a1fEXAMPLE",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
        "ConnectivityType": "private"
    }
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous CreateNatGatewayà la section Référence des AWS CLI commandes.

`create-network-acl-entry`

L'exemple de code suivant montre comment utiliser`create-network-acl-entry`.

AWS CLI

Pour créer une entrée ACL réseau

Cet exemple crée une entrée pour l'ACL réseau spécifiée. La règle autorise le trafic entrant depuis n'importe quelle IPv4 adresse (0.0.0.0/0) sur le port UDP 53 (DNS) vers n'importe quel sous-réseau associé. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 0.0.0.0/0 --rule-action allow
```

Cet exemple crée une règle pour l'ACL réseau spécifiée qui autorise le trafic entrant depuis n'importe quelle IPv6 adresse (: : /0) sur le port TCP 80 (HTTP).

Commande :

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 120 --protocol tcp --port-range From=80,To=80 --ipv6-cidr-block ::/0 --rule-action allow
```

- Pour plus de détails sur l'API, reportez-vous CreateNetworkAclEntryà la section Référence des AWS CLI commandes.

`create-network-acl`

L'exemple de code suivant montre comment utiliser`create-network-acl`.

AWS CLI

Pour créer une ACL réseau

Cet exemple crée une ACL réseau pour le VPC spécifié.

Commande :

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

Sortie :

```
{
    "NetworkAcl": {
        "Associations": [],
        "NetworkAclId": "acl-5fb85d36",
        "VpcId": "vpc-a01106c2",
        "Tags": [],
        "Entries": [
            {
                "CidrBlock": "0.0.0.0/0",
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": true,
                "RuleAction": "deny"
            },
            {
                "CidrBlock": "0.0.0.0/0",
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": false,
                "RuleAction": "deny"
            }
        ],
        "IsDefault": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateNetworkAclà la section Référence des AWS CLI commandes.

`create-network-insights-access-scope`

L'exemple de code suivant montre comment utiliser`create-network-insights-access-scope`.

AWS CLI

Pour créer une étendue d'accès réseau

L'`create-network-insights-access-scope`exemple suivant crée une étendue d'accès réseau.

```
aws ec2 create-network-insights-access-scope \
    --cli-input-json file://access-scope-file.json
```

Contenu de `access-scope-file.json` :

```
{
    "MatchPaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "Resources": [
                        "vpc-abcd12e3"
                    ]
                }
            }
         }
    ],
    "ExcludePaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "ResourceTypes": [
                        "AWS::EC2::InternetGateway"
                    ]
                }
            }
        }
    ]
}
```

Sortie :

```
{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789abc01234",
        "CreatedDate": "2022-01-25T19:20:28.796000+00:00",
        "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
    },
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
```

```
                        "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous CreateNetworkInsightsAccessScopeà la section Référence des AWS CLI commandes.

`create-network-insights-path`

L'exemple de code suivant montre comment utiliser`create-network-insights-path`.

AWS CLI

Pour créer un chemin

L'`create-network-insights-path`exemple suivant crée un chemin. La source est la passerelle Internet spécifiée et la destination est l' EC2 instance spécifiée. Pour déterminer si la destination est accessible via le protocole et le port spécifiés, analysez le chemin à l'aide de la `start-network-insights-analysis` commande.

```
aws ec2 create-network-insights-path \
    --source igw-0797cccdc9d73b0e5 \
```

```
    --destination i-0495d385ad28331c7 \
    --destination-port 22 \
    --protocol TCP
```

Sortie :

```
{
    "NetworkInsightsPaths": {
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
        "CreatedDate": "2021-01-20T22:43:46.933Z",
        "Source": "igw-0797cccdc9d73b0e5",
        "Destination": "i-0495d385ad28331c7",
        "Protocol": "tcp"
    }
}
```

Pour plus d'informations, consultez Getting started using the AWS CLI dans le guide Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous CreateNetworkInsightsPathà la section Référence des AWS CLI commandes.

`create-network-interface-permission`

L'exemple de code suivant montre comment utiliser`create-network-interface-permission`.

AWS CLI

Pour créer une autorisation d'interface réseau

Cet exemple autorise le compte `123456789012` à associer une interface réseau `eni-1a2b3c4d` à une instance.

Commande :

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

Sortie :

```
{
  "InterfacePermission": {
      "PermissionState": {
          "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-1a2b3c4d",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateNetworkInterfacePermissionà la section Référence des AWS CLI commandes.

`create-network-interface`

L'exemple de code suivant montre comment utiliser`create-network-interface`.

AWS CLI

Exemple 1 : pour spécifier une IPv4 adresse pour une interface réseau

L'`create-network-interface`exemple suivant crée une interface réseau pour le sous-réseau spécifié avec l' IPv4 adresse principale spécifiée.

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my network interface" \
    --groups sg-09dfba7ed20cda78b \
    --private-ip-address 10.0.8.17
```

Sortie :

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "my network interface",
        "Groups": [
            {
                "GroupName": "my-security-group",
                "GroupId": "sg-09dfba7ed20cda78b"
```

```
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [],
        "MacAddress": "06:6a:0f:9a:49:37",
        "NetworkInterfaceId": "eni-0492b355f0cf3b3f8",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.17",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-17.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.17"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b"
    }
}
```

Exemple 2 : pour créer une interface réseau avec une IPv4 adresse et une IPv6 adresse

L'`create-network-interface`exemple suivant crée une interface réseau pour le sous-réseau spécifié avec une IPv4 adresse et une IPv6 adresse sélectionnées par Amazon EC2.

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my dual stack network interface" \
    --ipv6-address-count 1 \
    --groups sg-09dfba7ed20cda78b
```

Sortie :

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "my dual stack network interface",
```

```
        "Groups": [
            {
                "GroupName": "my-security-group",
                "GroupId": "sg-09dfba7ed20cda78b"
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [
            {
                "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7",
                "IsPrimaryIpv6": false
            }
        ],
        "MacAddress": "06:b8:68:d2:b2:2d",
        "NetworkInterfaceId": "eni-05da417453f9a84bf",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.18",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.18"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b",
        "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7"
    }
}
```

Exemple 3 : pour créer une interface réseau avec des options de configuration de suivi des connexions

L'`create-network-interface`exemple suivant crée une interface réseau et configure les délais de suivi des connexions inactives.

```
aws ec2 create-network-interface \
```

```
    --subnet-id subnet-00a24d0d67acf6333 \
    --groups sg-02e57dbcfe0331c1b \
    --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60
```

Sortie :

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "ConnectionTrackingConfiguration": {
            "TcpEstablishedTimeout": 86400,
            "UdpTimeout": 60
        },
        "Description": "",
        "Groups": [
            {
                "GroupName": "my-security-group",
                "GroupId": "sg-02e57dbcfe0331c1b"
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [],
        "MacAddress": "06:4c:53:de:6d:91",
        "NetworkInterfaceId": "eni-0c133586e08903d0b",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.94",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.94"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b"
    }
}
```

Exemple 4 : pour créer un adaptateur Elastic Fabric

L'`create-network-interface`exemple suivant crée un EFA.

```
aws ec2 create-network-interface \
    --interface-type efa \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my efa" \
    --groups sg-02e57dbcfe0331c1b
```

Sortie :

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "my efa",
        "Groups": [
            {
                "GroupName": "my-efa-sg",
                "GroupId": "sg-02e57dbcfe0331c1b"
            }
        ],
        "InterfaceType": "efa",
        "Ipv6Addresses": [],
        "MacAddress": "06:d7:a4:f7:4d:57",
        "NetworkInterfaceId": "eni-034acc2885e862b65",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.180",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.180"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b"
```

```
    }
}
```

Pour plus d'informations, consultez la section Elastic network interfaces dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateNetworkInterfaceà la section Référence des AWS CLI commandes.

`create-placement-group`

L'exemple de code suivant montre comment utiliser`create-placement-group`.

AWS CLI

Pour créer un groupe de placement

Cet exemple de commande crée un groupe de placement portant le nom spécifié.

Commande :

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

Pour créer un groupe de placement de partitions

Cet exemple de commande crée un groupe de placement de partitions nommé `HDFS-Group-A` avec cinq partitions.

Commande :

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --
partition-count 5
```

- Pour plus de détails sur l'API, reportez-vous CreatePlacementGroupà la section Référence des AWS CLI commandes.

`create-replace-root-volume-task`

L'exemple de code suivant montre comment utiliser`create-replace-root-volume-task`.

AWS CLI

Exemple 1 : pour rétablir l'état de lancement initial d'un volume racine

L'`create-replace-root-volume-task`exemple suivant restaure le volume racine de l'instance i-0123456789abcdefa à son état de lancement initial.

```
aws ec2 create-replace-root-volume-task \
    --instance-id i-0123456789abcdefa
```

Sortie :

```
{
    "ReplaceRootVolumeTask":
    {
        "InstanceId": "i-0123456789abcdefa",
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "TaskState": "pending",
            "StartTime": "2022-03-14T15:06:38Z",
            "Tags": []
    }
}
```

Exemple 2 : pour restaurer un volume racine sur un instantané spécifique

L'`create-replace-root-volume-task`exemple suivant restaure le volume racine de l'instance i-0123456789abcdefa sur le snapshot snap-0abcdef1234567890.

```
aws ec2 create-replace-root-volume-task \
    --instance-id i-0123456789abcdefa \
    --snapshot-id  snap-0abcdef1234567890
```

Sortie :

```
{
    "ReplaceRootVolumeTask":
    {
        "InstanceId": "i-0123456789abcdefa",
        "ReplaceRootVolumeTaskId": "replacevol-0555566667777abcd",
        "TaskState": "pending",
        "StartTime": "2022-03-14T15:16:28Z",
```

```
        "Tags": []
    }
}
```

Pour plus d'informations, consultez la section Remplacer un volume racine dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateReplaceRootVolumeTaskà la section Référence des AWS CLI commandes.

`create-reserved-instances-listing`

L'exemple de code suivant montre comment utiliser`create-reserved-instances-listing`.

AWS CLI

Pour répertorier une instance réservée sur le Reserved Instance Marketplace

L'`create-reserved-instances-listing`exemple suivant crée une liste pour l'instance réservée spécifiée sur le Reserved Instance Marketplace.

```
aws ec2 create-reserved-instances-listing \
    --reserved-instances-id 5ec28771-05ff-4b9b-aa31-9e57dexample \
    --instance-count 3 \
    --price-schedules CurrencyCode=USD,Price=25.50 \
    --client-token 550e8400-e29b-41d4-a716-446655440000
```

- Pour plus de détails sur l'API, reportez-vous CreateReservedInstancesListingà la section Référence des AWS CLI commandes.

`create-restore-image-task`

L'exemple de code suivant montre comment utiliser`create-restore-image-task`.

AWS CLI

Pour restaurer une AMI à partir d'un compartiment S3

L'`create-restore-image-task`exemple suivant restaure une AMI à partir d'un compartiment S3. Utilisez les valeurs `S3ObjectKey `` and ``Bucket` issues de la `describe-store-`

`image-tasks` sortie, spécifiez la clé d'objet de l'AMI et le nom du compartiment S3 dans lequel l'AMI a été copiée, puis spécifiez le nom de l'AMI restaurée. Le nom de ce compte doit être unique AMIs dans la région. L'AMI restaurée recevra un nouvel ID d'AMI.

```
aws ec2 create-restore-image-task \
    --object-key ami-1234567890abcdef0.bin \
    --bucket my-ami-bucket \
    --name 'New AMI Name'
```

Sortie :

```
{
    "ImageId": "ami-0eab20fe36f83e1a8"
}
```

Pour plus d'informations, consultez Stocker et restaurer une AMI à l'aide de S3 dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateRestoreImageTaskà la section Référence des AWS CLI commandes.

`create-route-table`

L'exemple de code suivant montre comment utiliser`create-route-table`.

AWS CLI

Création d'une table de routage

Cet exemple crée une table de routage pour le VPC spécifié.

Commande :

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

Sortie :

```
{
    "RouteTable": {
```

```
        "Associations": [],
        "RouteTableId": "rtb-22574640",
        "VpcId": "vpc-a01106c2",
        "PropagatingVgws": [],
        "Tags": [],
        "Routes": [
            {
                "GatewayId": "local",
                "DestinationCidrBlock": "10.0.0.0/16",
                "State": "active"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRouteTableà la section Référence des AWS CLI commandes.

`create-route`

L'exemple de code suivant montre comment utiliser`create-route`.

AWS CLI

Pour créer un itinéraire

Cet exemple crée un itinéraire pour la table de routage spécifiée. L'itinéraire correspond à tout IPv4 le trafic (`0.0.0.0/0`) et l'achemine vers la passerelle Internet spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-
block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

Cet exemple de commande crée une route dans la table de routage rtb-g8ff4ea2. L'itinéraire fait correspondre le trafic du bloc IPv4 CIDR 10.0.0.0/16 et l'achemine vers la connexion d'appairage VPC, pcx-111aaa22. Cette route permet de diriger le trafic vers le VPC homologue dans la connexion d'appairage VPC. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-
block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

Cet exemple crée un itinéraire dans la table de routage spécifiée qui correspond à l'ensemble IPv6 du trafic (`::/0`) et l'achemine vers la passerelle Internet de sortie uniquement spécifiée.

Commande :

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-
block ::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- Pour plus de détails sur l'API, reportez-vous CreateRouteà la section Référence des AWS CLI commandes.

`create-security-group`

L'exemple de code suivant montre comment utiliser`create-security-group`.

AWS CLI

Pour créer un groupe de sécurité pour EC2 -Classic

Cet exemple crée un groupe de sécurité nommé `MySecurityGroup`.

Commande :

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My
 security group"
```

Sortie :

```
{
    "GroupId": "sg-903004f8"
}
```

Pour créer un groupe de sécurité pour EC2 -VPC

Cet exemple crée un groupe de sécurité nommé `MySecurityGroup` pour le VPC spécifié.

Commande :

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My
 security group" --vpc-id vpc-1a2b3c4d
```

Sortie :

```
{
    "GroupId": "sg-903004f8"
}
```

Pour plus d'informations, veuillez consulter la rubrique Using Security Groups dans le Guide de l'utilisateur AWS Command Line Interface.

- Pour plus de détails sur l'API, reportez-vous CreateSecurityGroupà la section Référence des AWS CLI commandes.

`create-snapshot`

L'exemple de code suivant montre comment utiliser`create-snapshot`.

AWS CLI

Pour créer un instantané

Cet exemple de commande crée un instantané du volume avec un ID de volume `vol-1234567890abcdef0` et une brève description pour identifier l'instantané.

Commande :

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my
 root volume snapshot"
```

Sortie :

```
{
    "Description": "This is my root volume snapshot",
    "Tags": [],
    "Encrypted": false,
```

```
    "VolumeId": "vol-1234567890abcdef0",
    "State": "pending",
    "VolumeSize": 8,
    "StartTime": "2018-02-28T21:06:01.000Z",
    "Progress": "",
    "OwnerId": "012345678910",
    "SnapshotId": "snap-066877671789bd71b"
}
```

Pour créer un instantané avec des balises

Cet exemple de commande crée un instantané et applique deux balises : purpose=prod et costcenter=123.

Commande :

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description 'Prod
 backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},
{Key=costcenter,Value=123}]'
```

Sortie :

```
{
    "Description": "Prod backup",
    "Tags": [
        {
            "Value": "prod",
            "Key": "purpose"
        },
        {
            "Value": "123",
            "Key": "costcenter"
        }
     ],
     "Encrypted": false,
     "VolumeId": "vol-1234567890abcdef0",
     "State": "pending",
     "VolumeSize": 8,
     "StartTime": "2018-02-28T21:06:06.000Z",
     "Progress": "",
     "OwnerId": "012345678910",
     "SnapshotId": "snap-09ed24a70bc19bbe4"
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSnapshotà la section Référence des AWS CLI commandes.

`create-snapshots`

L'exemple de code suivant montre comment utiliser`create-snapshots`.

AWS CLI

Exemple 1 : pour créer un instantané en plusieurs volumes

L'`create-snapshots`exemple suivant crée des instantanés de tous les volumes attachés à l'instance spécifiée.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --description "This is snapshot of a volume from my-instance"
```

Sortie :

```
{
    "Snapshots": [
        {
            "Description": "This is a snapshot of a volume from my-instance",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0a01d2d5a34697479",
            "State": "pending",
            "VolumeSize": 16,
            "StartTime": "2019-08-05T16:58:19.000Z",
            "Progress": "",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-07f30e3909aa0045e"
        },
        {
            "Description": "This is a snapshot of a volume from my-instance",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-02d0d4947008cb1a2",
            "State": "pending",
```

```
            "VolumeSize": 20,
            "StartTime": "2019-08-05T16:58:19.000Z",
            "Progress": "",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-0ec20b602264aad48"
        },
        ...
    ]
}
```

Exemple 2 : pour créer un instantané multivolume avec des balises provenant du volume source

L'`create-snapshots`exemple suivant crée des instantanés de tous les volumes attachés à l'instance spécifiée et copie les balises de chaque volume vers l'instantané correspondant.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --copy-tags-from-source volume \
    --description "This is snapshot of a volume from my-instance"
```

Sortie :

```
{
    "Snapshots": [
        {
            "Description": "This is a snapshot of a volume from my-instance",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-volume"
                }
            ],
            "Encrypted": false,
            "VolumeId": "vol-02d0d4947008cb1a2",
            "State": "pending",
            "VolumeSize": 20,
            "StartTime": "2019-08-05T16:53:04.000Z",
            "Progress": "",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-053bfaeb821a458dd"
        }
        ...
    ]
```

```
}
```

Exemple 3 : pour créer un instantané multivolume sans inclure le volume racine

L'`create-snapshots`exemple suivant crée un instantané de tous les volumes attachés à l'instance spécifiée, à l'exception du volume racine.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true
```

Voir l'exemple 1 pour un exemple de sortie.

Exemple 4 : pour créer un instantané en plusieurs volumes et ajouter des balises

L'`create-snapshots`exemple suivant crée des instantanés de tous les volumes attachés à l'instance spécifiée et ajoute deux balises à chaque instantané.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},
{Key=costcenter,Value=123}]'
```

Voir l'exemple 1 pour un exemple de sortie.

- Pour plus de détails sur l'API, reportez-vous CreateSnapshotsà la section Référence des AWS CLI commandes.

`create-spot-datafeed-subscription`

L'exemple de code suivant montre comment utiliser`create-spot-datafeed-subscription`.

AWS CLI

Pour créer un flux de données d'instance Spot

L'`create-spot-datafeed-subscription`exemple suivant crée un flux de données d'instance Spot.

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket \
```

```
    --prefix spot-data-feed
```

Sortie :

```
{
    "SpotDatafeedSubscription": {
        "Bucket": "amzn-s3-demo-bucket",
        "OwnerId": "123456789012",
        "Prefix": "spot-data-feed",
        "State": "Active"
    }
}
```

Le flux de données est stocké dans le compartiment Amazon S3 que vous avez spécifié. Les noms de fichiers de cette source de données sont au format suivant.

```
amzn-s3-demo-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-
HH.n.abcd1234.gz
```

Pour plus d'informations, consultez le flux de données des instances Spot dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateSpotDatafeedSubscriptionà la section Référence des AWS CLI commandes.

`create-store-image-task`

L'exemple de code suivant montre comment utiliser`create-store-image-task`.

AWS CLI

Pour stocker une AMI dans un compartiment S3

L'`create-store-image-task`exemple suivant enregistre une AMI dans un compartiment S3. Spécifiez l'ID de l'AMI et le nom du compartiment S3 dans lequel stocker l'AMI.

```
aws ec2 create-store-image-task \
  --image-id ami-1234567890abcdef0 \
  --bucket my-ami-bucket
```

Sortie :

```
{
    "ObjectKey": "ami-1234567890abcdef0.bin"
}
```

Pour plus d'informations, consultez Stocker et restaurer une AMI à l'aide de S3 dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateStoreImageTaskà la section Référence des AWS CLI commandes.

`create-subnet-cidr-reservation`

L'exemple de code suivant montre comment utiliser`create-subnet-cidr-reservation`.

AWS CLI

Pour créer une réservation CIDR de sous-réseau

L'`create-subnet-cidr-reservation`exemple suivant crée une réservation d'adresse CIDR de sous-réseau pour le sous-réseau et la plage d'adresses CIDR spécifiés.

```
aws ec2 create-subnet-cidr-reservation \
    --subnet-id subnet-03c51e2eEXAMPLE \
    --reservation-type prefix \
    --cidr 10.1.0.20/26
```

Sortie :

```
{
    "SubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

Pour plus d’informations, consultez Subnet CIDR reservations dans le Guide de l’utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous CreateSubnetCidrReservationà la section Référence des AWS CLI commandes.

`create-subnet`

L'exemple de code suivant montre comment utiliser`create-subnet`.

AWS CLI

Exemple 1 : pour créer un sous-réseau avec un bloc IPv4 CIDR uniquement

L'`create-subnet`exemple suivant crée un sous-réseau dans le VPC spécifié avec le bloc CIDR IPv4 spécifié.

```
aws ec2 create-subnet \
    --vpc-id vpc-081ec835f3EXAMPLE \
    --cidr-block 10.0.0.0/24 \
    --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

Sortie :

```
{
    "Subnet": {
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az2",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.0.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-0e99b93155EXAMPLE",
        "VpcId": "vpc-081ec835f3EXAMPLE",
        "OwnerId": "123456789012",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-ipv4-only-subnet"
            }
        ],
```

```
        "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0e99b93155EXAMPLE"
    }
}
```

Exemple 2 : pour créer un sous-réseau avec à la fois des blocs IPv6 CIDR IPv4 et des blocs CIDR

L'`create-subnet`exemple suivant crée un sous-réseau dans le VPC spécifié avec les blocs CIDR IPv6 et CIDR IPv4 spécifiés.

```
aws ec2 create-subnet \
    --vpc-id vpc-081ec835f3EXAMPLE \
    --cidr-block 10.0.0.0/24 \
    --ipv6-cidr-block 2600:1f16:cfe:3660::/64 \
    --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-ipv6-
subnet}]
```

Sortie :

```
{
    "Subnet": {
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az2",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.0.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-0736441d38EXAMPLE",
        "VpcId": "vpc-081ec835f3EXAMPLE",
        "OwnerId": "123456789012",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "subnet-cidr-assoc-06c5f904499fcc623",
                "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
                "Ipv6CidrBlockState": {
                    "State": "associating"
                }
            }
        ],
        "Tags": [
            {
```

```
                "Key": "Name",
                "Value": "my-ipv4-ipv6-subnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0736441d38EXAMPLE"
    }
}
```

Exemple 3 : pour créer un sous-réseau avec un bloc IPv6 CIDR uniquement

L'`create-subnet`exemple suivant crée un sous-réseau dans le VPC spécifié avec le bloc CIDR IPv6 spécifié.

```
aws ec2 create-subnet \
    --vpc-id vpc-081ec835f3EXAMPLE \
    --ipv6-native \
    --ipv6-cidr-block 2600:1f16:115:200::/64 \
    --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv6-only-
subnet}]
```

Sortie :

```
{
    "Subnet": {
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az2",
        "AvailableIpAddressCount": 0,
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-03f720e7deEXAMPLE",
        "VpcId": "vpc-081ec835f3EXAMPLE",
        "OwnerId": "123456789012",
        "AssignIpv6AddressOnCreation": true,
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "subnet-cidr-assoc-01ef639edde556709",
                "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
                "Ipv6CidrBlockState": {
                    "State": "associating"
                }
            }
```

```
        ],
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-ipv6-only-subnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-03f720e7deEXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section VPCs et les sous-réseaux dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous CreateSubnetà la section Référence des AWS CLI commandes.

`create-tags`

L'exemple de code suivant montre comment utiliser`create-tags`.

AWS CLI

Exemple 1 : pour ajouter une balise à une ressource

L'exemple `create-tags` suivant ajoute la balise `Stack=production` à l'image spécifiée ou remplace une balise existante pour l'AMI dont la clé de balise est `Stack`.

```
aws ec2 create-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=production
```

Cette commande ne produit aucune sortie

Exemple 2 : pour ajouter des balises à plusieurs ressources

L'exemple `create-tags` suivant ajoute (ou remplace) deux balises pour une AMI et une instance. L'une des balises contient une clé (`webserver`), mais pas de valeur (la valeur est définie comme une chaîne vide). L'autre balise contient une clé (`stack`) et une valeur (`Production`).

```
aws ec2 create-tags \
    --resources ami-1a2b3c4d i-1234567890abcdef0 \
    --tags Key=webserver,Value=   Key=stack,Value=Production
```

Cette commande ne produit aucune sortie

Exemple 3 : pour ajouter des balises contenant des caractères spéciaux

Les `create-tags` exemples suivants ajoutent la balise `[Group]=test` pour une instance. Les crochets ([et]) sont des caractères spéciaux, qui doivent être échappés. Les exemples suivants utilisent également le caractère de continuation de ligne approprié à chaque environnement.

Si vous utilisez Windows, entourez l'élément contenant des caractères spéciaux de guillemets doubles («), puis faites précéder chaque guillemet d'une barre oblique inverse (\) comme suit.

```
aws ec2 create-tags ^
    --resources i-1234567890abcdef0 ^
    --tags Key=\"[Group]\",Value=test
```

Si vous utilisez Windows PowerShell, entourez l'élément de la valeur comportant des caractères spéciaux de guillemets doubles («), faites précéder chaque guillemet d'une barre oblique inverse (\), puis entourez l'ensemble de la structure de clé et de valeur de guillemets simples (') comme suit.

```
aws ec2 create-tags `
    --resources i-1234567890abcdef0 `
    --tags 'Key=\"[Group]\",Value=test'
```

Si vous utilisez Linux ou OS X, entourez l'élément contenant des caractères spéciaux de guillemets doubles («), puis entourez l'ensemble de la structure des clés et des valeurs de guillemets simples (') comme suit.

```
aws ec2 create-tags \
    --resources i-1234567890abcdef0 \
    --tags 'Key="[Group]",Value=test'
```

Pour plus d'informations, consultez la section Marquer vos EC2 ressources Amazon dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateTagsà la section Référence des AWS CLI commandes.

`create-traffic-mirror-filter-rule`

L'exemple de code suivant montre comment utiliser`create-traffic-mirror-filter-rule`.

AWS CLI

Pour créer une règle de filtrage pour le trafic TCP entrant

L'`create-traffic-mirror-filter-rule`exemple suivant crée une règle que vous pouvez utiliser pour refléter tout le trafic TCP entrant. Avant d'exécuter cette commande, utilisez-la `create-traffic-mirror-filter` pour créer le filtre de miroir du trafic.

```
aws ec2 create-traffic-mirror-filter-rule \
    --description 'TCP Rule' \
    --destination-cidr-block 0.0.0.0/0  \
    --protocol 6 \
    --rule-action accept \
    --rule-number 1 \
    --source-cidr-block 0.0.0.0/0 \
    --traffic-direction ingress \
    --traffic-mirror-filter-id tmf-04812ff784b25ae67
```

Sortie :

```
{
    "TrafficMirrorFilterRule": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "TrafficMirrorFilterId": "tmf-04812ff784b25ae67",
        "TrafficMirrorFilterRuleId": "tmfr-02d20d996673f3732",
        "SourceCidrBlock": "0.0.0.0/0",
        "TrafficDirection": "ingress",
        "Description": "TCP Rule",
        "RuleNumber": 1,
        "RuleAction": "accept",
        "Protocol": 6
    },
    "ClientToken": "4752b573-40a6-4eac-a8a4-a72058761219"
}
```

Pour plus d'informations, voir Création d'un filtre de miroir du trafic dans le guide Traffic Mirroring.

- Pour plus de détails sur l'API, reportez-vous CreateTrafficMirrorFilterRuleà la section Référence des AWS CLI commandes.

`create-traffic-mirror-filter`

L'exemple de code suivant montre comment utiliser`create-traffic-mirror-filter`.

AWS CLI

Pour créer un filtre de rétroviseur du trafic

L'`create-traffic-mirror-filter`exemple suivant crée un filtre de miroir du trafic. Après avoir créé le filtre, utilisez-le `create-traffic-mirror-filter-rule` pour ajouter des règles.

```
aws ec2 create-traffic-mirror-filter \
    --description 'TCP Filter'
```

Sortie :

```
{
    "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",
    "TrafficMirrorFilter": {
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "Description": "TCP Filter",
        "EgressFilterRules": [],
        "IngressFilterRules": [],
        "Tags": [],
        "NetworkServices": []
    }
}
```

Pour plus d'informations, voir Création d'un filtre de miroir du trafic dans le guide Traffic Mirroring.

- Pour plus de détails sur l'API, reportez-vous CreateTrafficMirrorFilterà la section Référence des AWS CLI commandes.

`create-traffic-mirror-session`

L'exemple de code suivant montre comment utiliser`create-traffic-mirror-session`.

AWS CLI

Pour créer une session de miroir du trafic

La `create-traffic-mirror-session` commande suivante crée une session miroir de trafic pour la source et la cible spécifiées pour 25 octets du paquet.

```
aws ec2 create-traffic-mirror-session \
    --description 'example session' \
    --traffic-mirror-target-id tmt-07f75d8feeEXAMPLE \
    --network-interface-id eni-070203f901EXAMPLE \
    --session-number 1  \
    --packet-length 25 \
    --traffic-mirror-filter-id tmf-04812ff784EXAMPLE
```

Sortie :

```
{
    "TrafficMirrorSession": {
        "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
        "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "NetworkInterfaceId": "eni-070203f901EXAMPLE",
        "OwnerId": "111122223333",
        "PacketLength": 25,
        "SessionNumber": 1,
        "VirtualNetworkId": 7159709,
        "Description": "example session",
        "Tags": []
    },
    "ClientToken": "5236cffc-ee13-4a32-bb5b-388d9da09d96"
}
```

Pour plus d'informations, voir Création d'une session de miroir du trafic dans le Guide de mise en miroir du trafic.

- Pour plus de détails sur l'API, reportez-vous CreateTrafficMirrorSessionà la section Référence des AWS CLI commandes.

`create-traffic-mirror-target`

L'exemple de code suivant montre comment utiliser`create-traffic-mirror-target`.

AWS CLI

Pour créer une cible miroir de trafic Network Load Balancer

L'`create-traffic-mirror-target`exemple suivant crée une cible miroir de trafic Network Load Balancer.

```
aws ec2 create-traffic-mirror-target \
    --description 'Example Network Load Balancer Target' \
    --network-load-balancer-arn arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/net/NLB/7cdec873EXAMPLE
```

Sortie :

```
{
    "TrafficMirrorTarget": {
        "Type": "network-load-balancer",
        "Tags": [],
        "Description": "Example Network Load Balancer Target",
        "OwnerId": "111122223333",
        "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:724145273726:loadbalancer/net/NLB/7cdec873EXAMPLE",
        "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE"
    },
    "ClientToken": "d5c090f5-8a0f-49c7-8281-72c796a21f72"
}
```

Pour créer une cible miroir du trafic réseau

L'`create-traffic-mirror-target`exemple suivant crée une cible Traffic Mirror d'interface réseau.

```
aws ec2 create-traffic-mirror-target \
    --description 'Network interface target' \
    --network-interface-id eni-eni-01f6f631eEXAMPLE
```

Sortie :

```
{
    "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",
    "TrafficMirrorTarget": {
        "Description": "Network interface target",
```

```
        "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",
        "TrafficMirrorTargetId": "tmt-02dcdbe2abEXAMPLE",
        "OwnerId": "111122223333",
        "Type": "network-interface",
        "Tags": []
    }
}
```

Pour plus d'informations, voir Création d'une cible miroir du trafic dans le Guide de mise en miroir du trafic.

- Pour plus de détails sur l'API, reportez-vous CreateTrafficMirrorTargetà la section Référence des AWS CLI commandes.

`create-transit-gateway-connect-peer`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-connect-peer`.

AWS CLI

Pour créer un homologue Transit Gateway Connect

L'`create-transit-gateway-connect-peer`exemple suivant crée un pair Connect.

```
aws ec2 create-transit-gateway-connect-peer \
    --transit-gateway-attachment-id tgw-attach-0f0927767cEXAMPLE \
    --peer-address 172.31.1.11 \
    --inside-cidr-blocks 169.254.6.0/29
```

Sortie :

```
{
    "TransitGatewayConnectPeer": {
        "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
        "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
        "State": "pending",
        "CreationTime": "2021-10-13T03:35:17.000Z",
        "ConnectPeerConfiguration": {
            "TransitGatewayAddress": "10.0.0.234",
            "PeerAddress": "172.31.1.11",
            "InsideCidrBlocks": [
                "169.254.6.0/29"
```

```
            ],
            "Protocol": "gre",
            "BgpConfigurations": [
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.2",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                },
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.3",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez les pièces jointes Transit Gateway Connect et les homologues Transit Gateway Connect dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayConnectPeerà la section Référence des AWS CLI commandes.

`create-transit-gateway-connect`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-connect`.

AWS CLI

Pour créer une pièce jointe Connect pour une passerelle de transit

L'`create-transit-gateway-connect`exemple suivant crée une pièce jointe Connect, avec le protocole « gre », pour la pièce jointe spécifiée.

```
aws ec2 create-transit-gateway-connect \
    --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
    --options "Protocol=gre"
```

Sortie :

```
{
    "TransitGatewayConnect": {
        "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
        "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "pending",
        "CreationTime": "2021-03-09T19:59:17+00:00",
        "Options": {
            "Protocol": "gre"
        }
    }
}
```

Pour plus d'informations, consultez les pièces jointes Transit Gateway Connect et les homologues Transit Gateway Connect dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayConnectà la section Référence des AWS CLI commandes.

`create-transit-gateway-multicast-domain`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-multicast-domain`.

AWS CLI

Exemple 1 : pour créer un domaine de multidiffusion IGMP

L'`create-transit-gateway-multicast-domain`exemple suivant crée un domaine de multidiffusion pour la passerelle de transit spécifiée. Lorsque les sources statiques sont désactivées, toutes les instances des sous-réseaux associés au domaine de multidiffusion peuvent envoyer du trafic de multidiffusion. Si au moins un membre utilise le protocole IGMP, vous devez activer le IGMPv2 support.

```
aws ec2 create-transit-gateway-multicast-domain \
    --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
    --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

Sortie :

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",
        "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
        "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-0c9e29e2a7EXAMPLE",
        "OwnerId": "123456789012",
        "Options": {
            "Igmpv2Support": "enable",
            "StaticSourcesSupport": "disable",
            "AutoAcceptSharedAssociations": "disable"
        },
        "State": "pending",
        "CreationTime": "2021-09-29T22:17:13.000Z"
    }
}
```

Exemple 2 : pour créer un domaine de multidiffusion statique

L'`create-transit-gateway-multicast-domain`exemple suivant crée un domaine de multidiffusion pour la passerelle de transit spécifiée. Lorsque les sources statiques sont activées, vous devez ajouter des sources de manière statique.

```
aws ec2 create-transit-gateway-multicast-domain \
    --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
    --options StaticSourcesSupport=enable,Igmpv2Support=disable
```

Sortie :

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
        "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
        "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
        "OwnerId": "123456789012",
        "Options": {
            "Igmpv2Support": "disable",
            "StaticSourcesSupport": "enable",
            "AutoAcceptSharedAssociations": "disable"
```

```
        },
        "State": "pending",
        "CreationTime": "2021-09-29T22:20:19.000Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des domaines de multidiffusion dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayMulticastDomainà la section Référence des AWS CLI commandes.

`create-transit-gateway-peering-attachment`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-peering-attachment`.

AWS CLI

Pour créer une pièce jointe de peering pour une passerelle de transit

L'`create-transit-gateway-peering-attachment`exemple suivant crée une demande d'attachement d'appairage entre les deux passerelles de transit spécifiées.

```
aws ec2 create-transit-gateway-peering-attachment \
    --transit-gateway-id tgw-123abc05e04123abc \
    --peer-transit-gateway-id tgw-11223344aabbcc112 \
    --peer-account-id 123456789012 \
    --peer-region us-east-2
```

Sortie :

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
```

```
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "initiatingRequest",
        "CreationTime": "2019-12-09T11:38:05.000Z"
    }
}
```

Pour plus d'informations, consultez la section Transit Gateway peering Attachments dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayPeeringAttachmentà la section Référence des AWS CLI commandes.

`create-transit-gateway-policy-table`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-policy-table`.

AWS CLI

Pour créer un tableau des politiques relatives aux passerelles de transit

L'`create-transit-gateway-policy-table`exemple suivant crée une table de politique de passerelle de transit pour la passerelle de transit spécifiée.

```
aws ec2 create-transit-gateway-policy-table \
    --transit-gateway-id tgw-067f8505c18f0bd6e
```

Sortie :

```
{
    "TransitGatewayPolicyTable": {
        "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
        "TransitGatewayId": "tgw-067f8505c18f0bd6e",
        "State": "pending",
        "CreationTime": "2023-11-28T16:36:43+00:00"
    }
}
```

Pour plus d'informations, consultez les tableaux des politiques relatives aux passerelles de transit dans le guide de l'utilisateur de Transit Gateway.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayPolicyTableà la section Référence des AWS CLI commandes.

`create-transit-gateway-prefix-list-reference`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-prefix-list-reference`.

AWS CLI

Pour créer une référence à une liste de préfixes

L'`create-transit-gateway-prefix-list-reference`exemple suivant crée une référence à la liste de préfixes spécifiée dans la table de routage de la passerelle de transit spécifiée.

```
aws ec2 create-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-1111122222222333 \
    --transit-gateway-attachment-id tgw-attach-aaaaaabbbbbb11111
```

Sortie :

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "pending",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aaaaaabbbbbb11111",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

Pour plus d'informations, voir Création d'une liste de préfixes de référence dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayPrefixListReferenceà la section Référence des AWS CLI commandes.

`create-transit-gateway-route-table`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-route-table`.

AWS CLI

Pour créer une table de routage Transit Gateway

L'`create-transit-gateway-route-table`exemple suivant crée une table de routage pour la passerelle de transit spécifiée.

```
aws ec2 create-transit-gateway-route-table \
    --transit-gateway-id tgw-0262a0e521EXAMPLE
```

Sortie :

```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "State": "pending",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-10T19:01:46.000Z"
    }
}
```

Pour plus d'informations, voir Création d'une table de routage de passerelle de transit dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayRouteTableà la section Référence des AWS CLI commandes.

`create-transit-gateway-route`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-route`.

AWS CLI

Pour créer un itinéraire de passerelle de transit

L'`create-transit-gateway-route`exemple suivant crée un itinéraire, avec la destination spécifiée, pour la table de routage spécifiée.

```
aws ec2 create-transit-gateway-route \
    --destination-cidr-block 10.0.2.0/24 \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \
    --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Sortie :

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-0065acced4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "active"
    }
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayRouteà la section Référence des AWS CLI commandes.

`create-transit-gateway-vpc-attachment`

L'exemple de code suivant montre comment utiliser`create-transit-gateway-vpc-attachment`.

AWS CLI

Exemple 1 : pour associer une passerelle de transit à un VPC

L'`create-transit-gateway-vpc-attachment`exemple suivant crée une attache de passerelle de transit au VPC spécifié.

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-0262a0e521EXAMPLE \
```

```
    --vpc-id vpc-07e8ffd50f49335df \
    --subnet-id subnet-0752213d59EXAMPLE
```

Sortie :

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

Pour plus d'informations, voir Créer une passerelle de transit attachée à un VPC dans le Guide des passerelles de transit.

Exemple 2 : pour associer une passerelle de transit à plusieurs sous-réseaux dans un VPC

L'`create-transit-gateway-vpc-attachment`exemple suivant crée une attache de passerelle de transit au VPC et aux sous-réseaux spécifiés.

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-02f776b1a7EXAMPLE  \
    --vpc-id vpc-3EXAMPLE \
    --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

Sortie :

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0e141e0bebEXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
```

```
        "VpcId": "vpc-3EXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-6EXAMPLE",
            "subnet-dEXAMPLE"
        ],
        "CreationTime": "2019-12-17T20:07:52.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

Pour plus d'informations, voir Créer une passerelle de transit attachée à un VPC dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayVpcAttachmentà la section Référence des AWS CLI commandes.

`create-transit-gateway`

L'exemple de code suivant montre comment utiliser`create-transit-gateway`.

AWS CLI

Pour créer une passerelle de transit

L'`create-transit-gateway`exemple suivant crée une passerelle de transit.

```
aws ec2 create-transit-gateway \
    --description MyTGW \
    --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=
```

Sortie :

```
{
    "TransitGateway": {
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-gateway/
tgw-0262a0e521EXAMPLE",
```

```
        "State": "pending",
        "OwnerId": "111122223333",
        "Description": "MyTGW",
        "CreationTime": "2019-07-10T14:02:12.000Z",
        "Options": {
            "AmazonSideAsn": 64516,
            "AutoAcceptSharedAttachments": "enable",
            "DefaultRouteTableAssociation": "enable",
            "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
            "DefaultRouteTablePropagation": "enable",
            "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
            "VpnEcmpSupport": "enable",
            "DnsSupport": "enable"
        }
    }
}
```

Pour plus d'informations, voir Créer une passerelle de transit dans le Guide des passerelles de transport en commun.

- Pour plus de détails sur l'API, reportez-vous CreateTransitGatewayà la section Référence des AWS CLI commandes.

`create-verified-access-endpoint`

L'exemple de code suivant montre comment utiliser`create-verified-access-endpoint`.

AWS CLI

Pour créer un point de terminaison d'accès vérifié

L'`create-verified-access-endpoint`exemple suivant crée un point de terminaison Verified Access pour le groupe Verified Access spécifié. L'interface réseau et le groupe de sécurité spécifiés doivent appartenir au même VPC.

```
aws ec2 create-verified-access-endpoint \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --endpoint-type network-interface \
    --attachment-type vpc \
    --domain-certificate-arn arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE \
    --application-domain example.com \
    --endpoint-domain-prefix my-ava-app \
```

```
    --security-group-ids sg-004915970c4c8f13a \
    --network-interface-
options NetworkInterfaceId=eni-0aec70418c8d87a0f,Protocol=https,Port=443 \
    --tag-specifications ResourceType=verified-access-
endpoint,Tags=[{Key=Name,Value=my-va-endpoint}]
```

Sortie :

```
{
    "VerifiedAccessEndpoint": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
        "ApplicationDomain": "example.com",
        "EndpointType": "network-interface",
        "AttachmentType": "vpc",
        "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
        "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
        "SecurityGroupIds": [
            "sg-004915970c4c8f13a"
        ],
        "NetworkInterfaceOptions": {
            "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
            "Protocol": "https",
            "Port": 443
        },
        "Status": {
            "Code": "pending"
        },
        "Description": "",
        "CreationTime": "2023-08-25T20:54:43",
        "LastUpdatedTime": "2023-08-25T20:54:43",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-endpoint"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Points de terminaison Verified Access dans le Guide de l'utilisateur AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous CreateVerifiedAccessEndpointà la section Référence des AWS CLI commandes.

`create-verified-access-group`

L'exemple de code suivant montre comment utiliser`create-verified-access-group`.

AWS CLI

Pour créer un groupe d'accès vérifié

L'`create-verified-access-group`exemple suivant crée un groupe Verified Access pour l'instance Verified Access spécifiée.

```
aws ec2 create-verified-access-group \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --tag-specifications ResourceType=verified-access-
group,Tags=[{Key=Name,Value=my-va-group}]
```

Sortie :

```
{
    "VerifiedAccessGroup": {
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "",
        "Owner": "123456789012",
        "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
        "CreationTime": "2023-08-25T19:55:19",
        "LastUpdatedTime": "2023-08-25T19:55:19",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-group"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Groupes d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous CreateVerifiedAccessGroupà la section Référence des AWS CLI commandes.

`create-verified-access-instance`

L'exemple de code suivant montre comment utiliser`create-verified-access-instance`.

AWS CLI

Pour créer une instance d'accès vérifié

L'`create-verified-access-instance`exemple suivant crée une instance Verified Access avec une balise Name.

```
aws ec2 create-verified-access-instance \
    --tag-specifications ResourceType=verified-access-
instance,Tags=[{Key=Name,Value=my-va-instance}]
```

Sortie :

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T18:27:56",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-instance"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Instances d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous CreateVerifiedAccessInstanceà la section Référence des AWS CLI commandes.

`create-verified-access-trust-provider`

L'exemple de code suivant montre comment utiliser`create-verified-access-trust-provider`.

AWS CLI

Pour créer un fournisseur de confiance Verified Access

L'`create-verified-access-trust-provider`exemple suivant configure un fournisseur de confiance Verified Access à l'aide AWS d'Identity Center.

```
aws ec2 create-verified-access-trust-provider \
    --trust-provider-type user \
    --user-trust-provider-type iam-identity-center \
    --policy-reference-name idc \
    --tag-specifications ResourceType=verified-access-trust-
provider,Tags=[{Key=Name,Value=my-va-trust-provider}]
```

Sortie :

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T18:40:36",
        "LastUpdatedTime": "2023-08-25T18:40:36",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-trust-provider"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Trust providers for Verified Access dans le Guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous CreateVerifiedAccessTrustProviderà la section Référence des AWS CLI commandes.

`create-volume`

L'exemple de code suivant montre comment utiliser`create-volume`.

AWS CLI

Pour créer un volume SSD à usage général (gp2) vide

L'`create-volume`exemple suivant crée un volume SSD à usage général (gp2) de 80 GiB dans la zone de disponibilité spécifiée. Notez que la région actuelle doit l'êtreus`-east-1`, ou vous pouvez ajouter le `--region` paramètre pour spécifier la région pour la commande.

```
aws ec2 create-volume \
    --volume-type gp2 \
    --size 80 \
    --availability-zone us-east-1a
```

Sortie :

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 240,
    "SnapshotId": "",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 80
}
```

Si vous ne spécifiez aucun type de volume, le type de volume par défaut estgp2.

```
aws ec2 create-volume \
```

```
    --size 80 \
    --availability-zone us-east-1a
```

Exemple 2 : pour créer un volume SSD IOPS provisionné (io1) à partir d'un instantané

L'`create-volume`exemple suivant crée un volume SSD d'IOPS provisionnées (io1) avec 1 000 IOPS provisionnées dans la zone de disponibilité spécifiée à l'aide du snapshot spécifié.

```
aws ec2 create-volume \
    --volume-type io1 \
    --iops 1000 \
    --snapshot-id snap-066877671789bd71b \
    --availability-zone us-east-1a
```

Sortie :

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": false,
    "VolumeType": "io1",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 1000,
    "SnapshotId": "snap-066877671789bd71b",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 500
}
```

Exemple 3 : pour créer un volume chiffré

L'`create-volume`exemple suivant crée un volume chiffré à l'aide de la clé CMK par défaut pour le chiffrement EBS. Si le chiffrement est désactivé par défaut, vous devez spécifier le `--encrypted` paramètre comme suit.

```
aws ec2 create-volume \
    --size 80 \
    --encrypted \
    --availability-zone us-east-1a
```

Sortie :

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": true,
    "VolumeType": "gp2",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 240,
    "SnapshotId": "",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 80
}
```

Si le chiffrement est activé par défaut, l'exemple de commande suivant crée un volume chiffré, même sans le `--encrypted` paramètre.

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

Si vous utilisez le `--kms-key-id` paramètre pour spécifier une clé CMK gérée par le client, vous devez le `--encrypted` spécifier même si le chiffrement est activé par défaut.

```
aws ec2 create-volume \
    --volume-type gp2 \
    --size 80 \
    --encrypted \
    --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
    --availability-zone us-east-1a
```

Exemple 4 : pour créer un volume avec des balises

L'`create-volume`exemple suivant crée un volume et ajoute deux balises.

```
aws ec2 create-volume \
    --availability-zone us-east-1a \
    --volume-type gp2 \
    --size 80 \
    --tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},
{Key=cost-center,Value=cc123}]'
```

- Pour plus de détails sur l'API, reportez-vous CreateVolumeà la section Référence des AWS CLI commandes.

`create-vpc-endpoint-connection-notification`

L'exemple de code suivant montre comment utiliser`create-vpc-endpoint-connection-notification`.

AWS CLI

Pour créer une notification de connexion au point de terminaison

Cet exemple crée une notification pour un service de point de terminaison spécifique qui vous avertit lorsque des points de terminaison d'interface se sont connectés à votre service et lorsque des points de terminaison ont été acceptés pour votre service.

Commande :

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-
arn arn:aws:sns:us-east-2:123456789012:VpceNotification --connection-
events Connect Accept --service-id vpce-svc-1237881c0d25a3abc
```

Sortie :

```
{
   "ConnectionNotification": {
       "ConnectionNotificationState": "Enabled",
       "ConnectionNotificationType": "Topic",
       "ServiceId": "vpce-svc-1237881c0d25a3abc",
       "ConnectionEvents": [
           "Accept",
           "Connect"
       ],
       "ConnectionNotificationId": "vpce-nfn-008776de7e03f5abc",
       "ConnectionNotificationArn": "arn:aws:sns:us-
east-2:123456789012:VpceNotification"
   }
 }
```

- Pour plus de détails sur l'API, reportez-vous CreateVpcEndpointConnectionNotificationà la section Référence des AWS CLI commandes.

`create-vpc-endpoint-service-configuration`

L'exemple de code suivant montre comment utiliser`create-vpc-endpoint-service-configuration`.

AWS CLI

Exemple 1 : pour créer une configuration de service de point de terminaison pour un point de terminaison d'interface

L'`create-vpc-endpoint-service-configuration`exemple suivant crée une configuration de service de point de terminaison VPC à l'aide du Network Load Balancer. `nlb-vpce` Cet exemple indique également que les demandes de connexion au service via un point de terminaison d'interface doivent être acceptées.

```
aws ec2 create-vpc-endpoint-service-configuration \
    --network-load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/net/nlb-vpce/e94221227f1ba532 \
    --acceptance-required
```

Sortie :

```
{
   "ServiceConfiguration": {
       "ServiceType": [
           {
               "ServiceType": "Interface"
           }
       ],
       "NetworkLoadBalancerArns": [
           "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/
nlb-vpce/e94221227f1ba532"
       ],
       "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-03d5ebb7d9579a2b3",
       "ServiceState": "Available",
       "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",
       "AcceptanceRequired": true,
       "AvailabilityZones": [
           "us-east-1d"
       ],
       "BaseEndpointDnsNames": [
           "vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com"
       ]
```

```
    }
}
```

Pour plus d'informations, consultez la section Création d'un service de point de terminaison dans le guide de AWS PrivateLink l'utilisateur.

Exemple 2 : pour créer une configuration de service de point de terminaison pour un point de terminaison Gateway Load Balancer

L'`create-vpc-endpoint-service-configuration`exemple suivant crée une configuration de service de point de terminaison VPC à l'aide du Gateway Load Balancer. `GWLBService` Les demandes de connexion au service via un point de terminaison Gateway Load Balancer sont automatiquement acceptées.

```
aws ec2 create-vpc-endpoint-service-configuration \
    --gateway-load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/gwy/GWLBService/123123123123abcc \
    --no-acceptance-required
```

Sortie :

```
{
    "ServiceConfiguration": {
        "ServiceType": [
            {
                "ServiceType": "GatewayLoadBalancer"
            }
        ],
        "ServiceId": "vpce-svc-123123a1c43abc123",
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
        "ServiceState": "Available",
        "AvailabilityZones": [
            "us-east-1d"
        ],
        "AcceptanceRequired": false,
        "ManagesVpcEndpoints": false,
        "GatewayLoadBalancerArns": [
            "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/gwy/
GWLBService/123123123123abcc"
        ]
    }
}
```

Pour plus d'informations, consultez la section Create a Gateway Load Balancer endpoint service dans le guide de l'AWS PrivateLink utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateVpcEndpointServiceConfigurationà la section Référence des AWS CLI commandes.

`create-vpc-endpoint`

L'exemple de code suivant montre comment utiliser`create-vpc-endpoint`.

AWS CLI

Exemple 1 : pour créer un point de terminaison de passerelle

L'`create-vpc-endpoint`exemple suivant crée un point de terminaison VPC de passerelle entre le VPC et Amazon `vpc-1a2b3c4d` S3 dans la `us-east-1` région, et associe une table `rtb-11aa22bb` de routage au point de terminaison.

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --service-name com.amazonaws.us-east-1.s3 \
    --route-table-ids rtb-11aa22bb
```

Sortie :

```
{
    "VpcEndpoint": {
        "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",
\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}",
        "VpcId": "vpc-1a2b3c4d",
        "State": "available",
        "ServiceName": "com.amazonaws.us-east-1.s3",
        "RouteTableIds": [
            "rtb-11aa22bb"
        ],
        "VpcEndpointId": "vpc-1a2b3c4d",
        "CreationTimestamp": "2015-05-15T09:40:50Z"
    }
}
```

Pour plus d'informations, voir Création d'un point de terminaison de passerelle dans le guide de AWS PrivateLink l'utilisateur.

Exemple 2 : pour créer un point de terminaison d'interface

L'`create-vpc-endpoint`exemple suivant crée un point de terminaison VPC d'interface entre VPC et Amazon `vpc-1a2b3c4d` S3 dans la région. `us-east-1` La commande crée le point de terminaison dans un sous-réseau`subnet-1a2b3c4d`, l'associe au groupe `sg-1a2b3c4d` de sécurité et ajoute une balise avec la clé « Service » et la valeur « S3 ».

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --vpc-endpoint-type Interface \
    --service-name com.amazonaws.us-east-1.s3 \
    --subnet-ids subnet-7b16de0c \
    --security-group-id sg-1a2b3c4d \
    --tag-specifications ResourceType=vpc-endpoint,Tags=[{Key=service,Value=S3}]
```

Sortie :

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-1a2b3c4d5e6f1a2b3",
        "VpcEndpointType": "Interface",
        "VpcId": "vpc-1a2b3c4d",
        "ServiceName": "com.amazonaws.us-east-1.s3",
        "State": "pending",
        "RouteTableIds": [],
        "SubnetIds": [
            "subnet-1a2b3c4d"
        ],
        "Groups": [
            {
                "GroupId": "sg-1a2b3c4d",
                "GroupName": "default"
            }
        ],
        "PrivateDnsEnabled": false,
        "RequesterManaged": false,
        "NetworkInterfaceIds": [
            "eni-0b16f0581c8ac6877"
        ],
        "DnsEntries": [
            {
                "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg.s3.us-
east-1.vpce.amazonaws.com",
```

```
                "HostedZoneId": "Z7HUB22UULQXV"
            },
            {
                "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg-us-east-1c.s3.us-
east-1.vpce.amazonaws.com",
                "HostedZoneId": "Z7HUB22UULQXV"
            }
        ],
        "CreationTimestamp": "2021-03-05T14:46:16.030000+00:00",
        "Tags": [
            {
                "Key": "service",
                "Value": "S3"
            }
        ],
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Créer un point de terminaison VPC d'interface dans le guide de l'AWS PrivateLink utilisateur.

Exemple 3 : pour créer un point de terminaison Gateway Load Balancer

L'`create-vpc-endpoint`exemple suivant crée un point de terminaison Gateway Load Balancer entre un VPC `vpc-111122223333aabbc` et un service configuré à l'aide d'un Gateway Load Balancer.

```
aws ec2 create-vpc-endpoint \
    --service-name com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123 \
    --vpc-endpoint-type GatewayLoadBalancer \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445
```

Sortie :

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-aabbaabbaabbaabba",
        "VpcEndpointType": "GatewayLoadBalancer",
        "VpcId": "vpc-111122223333aabbc",
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
```

```
        "State": "pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "RequesterManaged": false,
        "NetworkInterfaceIds": [
            "eni-01010120203030405"
        ],
        "CreationTimestamp": "2020-11-11T08:06:03.522Z",
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Points de terminaison Gateway Load Balancer dans le guide de l'AWS PrivateLink utilisateur.

Exemple 4 : pour créer un point de terminaison de ressource

L'`create-vpc-endpoint`exemple suivant crée un point de terminaison de ressource.

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type Resource \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445 \
    --resource-configuration-arn arn:aws:vpc-lattice-us-
east-1:123456789012:resourceconfiguration/rcfg-0123abcde98765432
```

Sortie :

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-00939a7ed9EXAMPLE",
        "VpcEndpointType": "Resource",
        "VpcId": "vpc-111122223333aabbc",
        "State": "Pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "Groups": [
            {
                "GroupId": "sg-03e2f15fbfc09b000",
                "GroupName": "default"
            }
```

```
        ],
        "IpAddressType": "IPV4",
        "PrivateDnsEnabled": false,
        "CreationTimestamp": "2025-02-06T23:38:49.525000+00:00",
        "Tags": [],
        "OwnerId": "123456789012",
        "ResourceConfigurationArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123abcde98765432"
    }
}
```

Pour plus d'informations, consultez la section Resource endpoints dans le guide de l'AWS PrivateLink utilisateur.

Exemple 5 : pour créer un point de terminaison d'un réseau de services

L'`create-vpc-endpoint`exemple suivant crée un point de terminaison de réseau de services.

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type ServiceNetwork \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445 \
    --service-network-arn arn:aws:vpc-lattice:us-east-1:123456789012:servicenetwork/
sn-0101abcd5432abcd0 \
    --security-group-ids sg-0123456789012abcd
```

Sortie :

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-0f00567fa8EXAMPLE",
        "VpcEndpointType": "ServiceNetwork",
        "VpcId": "vpc-111122223333aabbc",
        "State": "Pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "Groups": [
            {
                "GroupId": "sg-0123456789012abcd",
                "GroupName": "my-security-group"
            }
        ],
```

```
        "IpAddressType": "IPV4",
        "PrivateDnsEnabled": false,
        "CreationTimestamp": "2025-02-06T23:44:20.449000+00:00",
        "Tags": [],
        "OwnerId": "123456789012",
        "ServiceNetworkArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:servicenetwork/sn-0101abcd5432abcd0"
    }
}
```

Pour plus d'informations, consultez la section Points de terminaison du réseau de service dans le Guide de AWS PrivateLink l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateVpcEndpointà la section Référence des AWS CLI commandes.

`create-vpc-peering-connection`

L'exemple de code suivant montre comment utiliser`create-vpc-peering-connection`.

AWS CLI

Pour créer une connexion d'appairage VPC entre votre VPCs

Cet exemple demande une connexion d'appairage entre votre VPCs vpc-1a2b3c4d et vpc-11122233.

Commande :

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233
```

Sortie :

```
{
    "VpcPeeringConnection": {
        "Status": {
            "Message": "Initiating Request to 444455556666",
            "Code": "initiating-request"
        },
        "Tags": [],
        "RequesterVpcInfo": {
```

```
            "OwnerId": "444455556666",
            "VpcId": "vpc-1a2b3c4d",
            "CidrBlock": "10.0.0.0/28"
        },
        "VpcPeeringConnectionId": "pcx-111aaa111",
        "ExpirationTime": "2014-04-02T16:13:36.000Z",
        "AccepterVpcInfo": {
            "OwnerId": "444455556666",
            "VpcId": "vpc-11122233"
        }
    }
}
```

Pour créer une connexion d'appairage VPC avec un VPC d'un autre compte

Cet exemple demande une connexion d'appairage entre votre VPC (vpc-1a2b3c4d) et un VPC (vpc-11122233) qui appartient au compte 123456789012. AWS

Commande :

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012
```

Pour créer une connexion d'appairage VPC avec un VPC d'une autre région

Cet exemple demande une connexion d'appairage entre votre VPC de la région actuelle (vpc-1a2b3c4d) et un VPC (vpc-11122233) de votre compte dans la région. `us-west-2`

Commande :

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-region us-west-2
```

Cet exemple demande une connexion d'appairage entre votre VPC de la région actuelle (vpc-1a2b3c4d) et un VPC (vpc-11122233) appartenant au compte 123456789012 situé dans la région. AWS `us-west-2`

Commande :

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2
```

- Pour plus de détails sur l'API, reportez-vous CreateVpcPeeringConnectionà la section Référence des AWS CLI commandes.

`create-vpc`

L'exemple de code suivant montre comment utiliser`create-vpc`.

AWS CLI

Exemple 1 : création d'un VPC

L'`create-vpc`exemple suivant crée un VPC avec le bloc IPv4 CIDR spécifié et une balise Name.

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

Sortie :

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-5EXAMPLE",
        "State": "pending",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-07501b79ecEXAMPLE",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "Tags": [
            {
                "Key": "Name",
                "Value": MyVpc"
            }
```

```
        ]
    }
}
```

Exemple 2 : création d'un VPC avec une location dédiée

L'`create-vpc`exemple suivant crée un VPC avec le bloc IPv4 CIDR spécifié et une location dédiée.

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --instance-tenancy dedicated
```

Sortie :

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-19edf471",
        "State": "pending",
        "VpcId": "vpc-0a53287fa4EXAMPLE",
        "OwnerId": "111122223333",
        "InstanceTenancy": "dedicated",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-00b24cc1c2EXAMPLE",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false
    }
}
```

Exemple 3 : pour créer un VPC avec un IPv6 bloc CIDR

L'`create-vpc`exemple suivant crée un VPC avec un bloc CIDR fourni par Amazon IPv6 .

```
aws ec2 create-vpc \
```

```
    --cidr-block 10.0.0.0/16 \
    --amazon-provided-ipv6-cidr-block
```

Sortie :

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-dEXAMPLE",
        "State": "pending",
        "VpcId": "vpc-0fc5e3406bEXAMPLE",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",
                "Ipv6CidrBlock": "",
                "Ipv6CidrBlockState": {
                    "State": "associating"
                },
                "Ipv6Pool": "Amazon",
                "NetworkBorderGroup": "us-west-2"
            }
        ],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false
    }
}
```

Exemple 4 : création d'un VPC avec un CIDR depuis un groupe IPAM

L'exemple `create-vpc` suivant crée un VPC avec un CIDR à partir d'un groupe de gestionnaire d'adresses IP (IPAM) Amazon VPC (IPAM).

Linux et macOS :

```
aws ec2 create-vpc \
    --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 \
    --tag-specifications ResourceType=vpc,Tags='[{Key=Environment,Value="Preprod"},
{Key=Owner,Value="Build Team"}]'
```

Windows :

```
aws ec2 create-vpc ^
    --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 ^
    --tag-specifications ResourceType=vpc,Tags=[{Key=Environment,Value="Preprod"},
{Key=Owner,Value="Build Team"}]
```

Sortie :

```
{
    "Vpc": {
        "CidrBlock": "10.0.1.0/24",
        "DhcpOptionsId": "dopt-2afccf50",
        "State": "pending",
        "VpcId": "vpc-010e1791024eb0af9",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-0a77de1d803226d4b",
                "CidrBlock": "10.0.1.0/24",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "Tags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            },
            {
                "Key": "Owner",
                "Value": "Build Team"
            }
```

```
        ]
    }
}
```

Pour plus d'informations, veuillez consulter la rubrique Création d'un VPC qui utilise un CIDR de groupe IPAM dans le Guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous CreateVpcà la section Référence des AWS CLI commandes.

`create-vpn-connection-route`

L'exemple de code suivant montre comment utiliser`create-vpn-connection-route`.

AWS CLI

Pour créer un itinéraire statique pour une connexion VPN

Cet exemple crée un itinéraire statique pour la connexion VPN spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- Pour plus de détails sur l'API, reportez-vous CreateVpnConnectionRouteà la section Référence des AWS CLI commandes.

`create-vpn-connection`

L'exemple de code suivant montre comment utiliser`create-vpn-connection`.

AWS CLI

Exemple 1 : pour créer une connexion VPN avec un routage dynamique

L'`create-vpn-connection`exemple suivant crée une connexion VPN entre la passerelle privée virtuelle spécifiée et la passerelle client spécifiée, et applique des balises à la connexion VPN. La sortie inclut les informations de configuration de votre dispositif de passerelle client, au format XML.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --tag-specification 'ResourceType=vpn-connection,Tags=[{Key=Name,Value=BGP-
VPN}]'
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {},
                {}
            ]
        },
        "Routes": [],
        "Tags": [
             {
                "Key": "Name",
                "Value": "BGP-VPN"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section AWS Site-to-Site Fonctionnement du VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

Exemple 2 : pour créer une connexion VPN avec un routage statique

L'`create-vpn-connection`exemple suivant crée une connexion VPN entre la passerelle privée virtuelle spécifiée et la passerelle client spécifiée. Les options spécifient le routage statique. La sortie inclut les informations de configuration de votre dispositif de passerelle client, au format XML.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --options "{\"StaticRoutesOnly\":true}"
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": true,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {},
                {}
            ]
        },
        "Routes": [],
        "Tags": []
    }
}
```

Pour plus d'informations, consultez la section AWS Site-to-Site Fonctionnement du VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

Exemple 3 : pour créer une connexion VPN et spécifier votre propre CIDR interne et votre clé pré-partagée

L'`create-vpn-connection`exemple suivant crée une connexion VPN et spécifie le bloc CIDR de l'adresse IP interne et une clé pré-partagée personnalisée pour chaque tunnel. Les valeurs spécifiées sont renvoyées dans les `CustomerGatewayConfiguration` informations.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --options
 TunnelOptions='[{TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},
{TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {
                    "OutsideIpAddress": "203.0.113.3",
                    "TunnelInsideCidr": "169.254.12.0/30",
                    "PreSharedKey": "ExamplePreSharedKey1"
                },
                {
                    "OutsideIpAddress": "203.0.113.5",
                    "TunnelInsideCidr": "169.254.13.0/30",
                    "PreSharedKey": "ExamplePreSharedKey2"
                }
            ]
        },
        "Routes": [],
        "Tags": []
```

```
    }
}
```

Pour plus d'informations, consultez la section AWS Site-to-Site Fonctionnement du VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

Exemple 4 : pour créer une connexion VPN qui prend en charge IPv6 le trafic

L'`create-vpn-connection`exemple suivant crée une connexion VPN qui prend en charge le IPv6 trafic entre la passerelle de transit spécifiée et la passerelle client spécifiée. Les options de tunnel pour les deux tunnels spécifient qui AWS doit lancer la négociation IKE.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --transit-gateway-id tgw-12312312312312312 \
    --customer-gateway-id cgw-001122334455aabbc \
    --options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},
{StartupAction=start}]
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-11111111122222222",
        "TransitGatewayId": "tgw-12312312312312312",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv6NetworkCidr": "::/0",
            "RemoteIpv6NetworkCidr": "::/0",
            "TunnelInsideIpVersion": "ipv6",
            "TunnelOptions": [
                {
                    "OutsideIpAddress": "203.0.113.3",
                    "StartupAction": "start"
                },
                {
                    "OutsideIpAddress": "203.0.113.5",
```

```
                    "StartupAction": "start"
                }
            ]
        },
        "Routes": [],
        "Tags": []
    }
}
```

Pour plus d'informations, consultez la section AWS Site-to-Site Fonctionnement du VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

- Pour plus de détails sur l'API, reportez-vous CreateVpnConnectionà la section Référence des AWS CLI commandes.

`create-vpn-gateway`

L'exemple de code suivant montre comment utiliser`create-vpn-gateway`.

AWS CLI

Pour créer une passerelle privée virtuelle

Cet exemple crée une passerelle privée virtuelle.

Commande :

```
aws ec2 create-vpn-gateway --type ipsec.1
```

Sortie :

```
{
    "VpnGateway": {
        "AmazonSideAsn": 64512,
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-9a4cacf3",
        "VpcAttachments": []
    }
}
```

Pour créer une passerelle privée virtuelle avec un ASN spécifique côté Amazon

Cet exemple crée une passerelle privée virtuelle et spécifie le numéro de système autonome (ASN) pour le côté Amazon de la session BGP.

Commande :

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

Sortie :

```
{
    "VpnGateway": {
        "AmazonSideAsn": 65001,
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-9a4cacf3",
        "VpcAttachments": []
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateVpnGatewayà la section Référence des AWS CLI commandes.

`delete-carrier-gateway`

L'exemple de code suivant montre comment utiliser`delete-carrier-gateway`.

AWS CLI

Pour supprimer votre passerelle d'opérateur

L'`delete-carrier-gateway`exemple suivant supprime la passerelle de transport spécifiée.

```
aws ec2 delete-carrier-gateway \
    --carrier-gateway-id cagw-0465cdEXAMPLE1111
```

Sortie :

```
{
    "CarrierGateway": {
        "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
        "VpcId": "vpc-0c529aEXAMPLE1111",
        "State": "deleting",
```

```
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez Carrier Gateways dans le guide de l'utilisateur d'Amazon Virtual Private Cloud.

- Pour plus de détails sur l'API, reportez-vous DeleteCarrierGatewayà la section Référence des AWS CLI commandes.

`delete-client-vpn-endpoint`

L'exemple de code suivant montre comment utiliser`delete-client-vpn-endpoint`.

AWS CLI

Pour supprimer un point de terminaison Client VPN

L'`delete-client-vpn-endpoint`exemple suivant supprime le point de terminaison VPN du Client spécifié.

```
aws ec2 delete-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "Status": {
        "Code": "deleting"
    }
}
```

Pour plus d'informations, consultez la section Points de terminaison du VPN client dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DeleteClientVpnEndpointà la section Référence des AWS CLI commandes.

`delete-client-vpn-route`

L'exemple de code suivant montre comment utiliser`delete-client-vpn-route`.

AWS CLI

Pour supprimer un itinéraire pour un point de terminaison VPN client

L'`delete-client-vpn-route`exemple suivant supprime la `0.0.0.0/0` route pour le sous-réseau spécifié d'un point de terminaison Client VPN.

```
aws ec2 delete-client-vpn-route \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --destination-cidr-block 0.0.0.0/0 \
    --target-vpc-subnet-id subnet-0123456789abcabca
```

Sortie :

```
{
    "Status": {
        "Code": "deleting"
    }
}
```

Pour plus d'informations, consultez Routes dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DeleteClientVpnRouteà la section Référence des AWS CLI commandes.

`delete-coip-cidr`

L'exemple de code suivant montre comment utiliser`delete-coip-cidr`.

AWS CLI

Pour supprimer une plage d'adresses IP (CoIP) appartenant au client

L'`delete-coip-cidr`exemple suivant supprime la plage d'adresses CoIP spécifiée dans le pool CoIP spécifié.

```
aws ec2 delete-coip-cidr \
    --cidr 14.0.0.0/24 \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Sortie :

```
{
    "CoipCidr": {
        "Cidr": "14.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

Pour plus d'informations, voir Adresses IP appartenant au client dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DeleteCoipCidrà la section Référence des AWS CLI commandes.

`delete-coip-pool`

L'exemple de code suivant montre comment utiliser`delete-coip-pool`.

AWS CLI

Pour supprimer un pool d'adresses IP (CoIP) appartenant au client

L'`delete-coip-pool`exemple suivant supprime un pool d'adresses CoIP.

```
aws ec2 delete-coip-pool \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Sortie :

```
{
    "CoipPool": {
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
    }
}
```

Pour plus d'informations, voir Adresses IP appartenant au client dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DeleteCoipPoolà la section Référence des AWS CLI commandes.

`delete-customer-gateway`

L'exemple de code suivant montre comment utiliser`delete-customer-gateway`.

AWS CLI

Pour supprimer une passerelle client

Cet exemple supprime la passerelle client spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- Pour plus de détails sur l'API, reportez-vous DeleteCustomerGatewayà la section Référence des AWS CLI commandes.

`delete-dhcp-options`

L'exemple de code suivant montre comment utiliser`delete-dhcp-options`.

AWS CLI

Pour supprimer un ensemble d'options DHCP

Cet exemple supprime le jeu d'options DHCP spécifié. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- Pour plus de détails sur l'API, reportez-vous DeleteDhcpOptionsà la section Référence des AWS CLI commandes.

`delete-egress-only-internet-gateway`

L'exemple de code suivant montre comment utiliser`delete-egress-only-internet-gateway`.

AWS CLI

Pour supprimer une passerelle Internet de sortie uniquement

Cet exemple supprime la passerelle Internet de sortie uniquement spécifiée.

Commande :

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-
id eigw-01eadbd45ecd7943f
```

Sortie :

```
{
  "ReturnCode": true
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteEgressOnlyInternetGatewayà la section Référence des AWS CLI commandes.

`delete-fleets`

L'exemple de code suivant montre comment utiliser`delete-fleets`.

AWS CLI

Exemple 1 : pour supprimer une EC2 flotte et mettre fin aux instances associées

L'`delete-fleets`exemple suivant supprime le EC2 parc spécifié et met fin aux instances à la demande et aux instances ponctuelles associées.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --terminate-instances
```

Sortie :

```
{
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
        }
    ],
    "UnsuccessfulFleetDeletions": []
}
```

Pour plus d'informations, consultez Supprimer un EC2 parc dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

Exemple 2 : Supprimer une EC2 flotte sans mettre fin aux instances associées

L'`delete-fleets`exemple suivant supprime le EC2 parc spécifié sans mettre fin aux instances à la demande et aux instances ponctuelles associées.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --no-terminate-instances
```

Sortie :

```
{
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
        }
    ],
    "UnsuccessfulFleetDeletions": []
}
```

Pour plus d'informations, consultez Supprimer un EC2 parc dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DeleteFleetsà la section Référence des AWS CLI commandes.

`delete-flow-logs`

L'exemple de code suivant montre comment utiliser`delete-flow-logs`.

AWS CLI

Pour supprimer un journal de flux

L'`delete-flow-logs`exemple suivant supprime le journal de flux spécifié.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

Sortie :

```
{
    "Unsuccessful": []
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteFlowLogsà la section Référence des AWS CLI commandes.

`delete-fpga-image`

L'exemple de code suivant montre comment utiliser`delete-fpga-image`.

AWS CLI

Pour supprimer une image Amazon FPGA

Cet exemple supprime l'AFI spécifié.

Commande :

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

Sortie :

```
{
  "Return": true
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteFpgaImageà la section Référence des AWS CLI commandes.

`delete-instance-connect-endpoint`

L'exemple de code suivant montre comment utiliser`delete-instance-connect-endpoint`.

AWS CLI

Pour supprimer un point de terminaison EC2 Instance Connect

L'`delete-instance-connect-endpoint`exemple suivant supprime le point de terminaison EC2 Instance Connect spécifié.

```
aws ec2 delete-instance-connect-endpoint \
    --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

Sortie :

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

Pour plus d'informations, consultez Remove EC2 Instance Connect Endpoint dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteInstanceConnectEndpointà la section Référence des AWS CLI commandes.

`delete-instance-event-window`

L'exemple de code suivant montre comment utiliser`delete-instance-event-window`.

AWS CLI

Exemple 1 : pour supprimer une fenêtre d'événements

L'`delete-instance-event-window`exemple suivant supprime une fenêtre d'événements.

```
aws ec2 delete-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890
```

Sortie :

```
{
    "InstanceEventWindowState": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "State": "deleting"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 2 : Pour forcer la suppression d'une fenêtre d'événements

L'`delete-instance-event-window`exemple suivant montre comment forcer la suppression d'une fenêtre d'événements si celle-ci est actuellement associée à des cibles.

```
aws ec2 delete-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --force-delete
```

Sortie :

```
{
    "InstanceEventWindowState": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
```

```
        "State": "deleting"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteInstanceEventWindowà la section Référence des AWS CLI commandes.

`delete-internet-gateway`

L'exemple de code suivant montre comment utiliser`delete-internet-gateway`.

AWS CLI

Pour supprimer une passerelle Internet

L'`delete-internet-gateway`exemple suivant supprime la passerelle Internet spécifiée.

```
aws ec2 delete-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Passerelles Internet dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DeleteInternetGatewayà la section Référence des AWS CLI commandes.

`delete-ipam-pool`

L'exemple de code suivant montre comment utiliser`delete-ipam-pool`.

AWS CLI

Pour supprimer un pool IPAM

Dans cet exemple, vous êtes un administrateur délégué IPAM qui souhaite supprimer un pool IPAM dont vous n'avez plus besoin, mais qui est doté d'un CIDR. Vous ne pouvez pas supprimer

un pool s'il y a CIDRs été provisionné, sauf si vous utilisez `--cascade` cette option, vous allez donc utiliser`--cascade`.

Pour compléter cette demande :

Vous aurez besoin de l'ID du pool IPAM que vous pouvez obtenir avec describe-ipam-pools. `--region` Il doit s'agir de la région d'origine de l'IPAM.

L'`delete-ipam-pool`exemple suivant supprime un pool IPAM de votre AWS compte.

```
aws ec2 delete-ipam-pool \
    --ipam-pool-id ipam-pool-050c886a3ca41cd5b \
    --cascade \
    --region us-east-1
```

Sortie :

```
{
    "IpamPool": {
        "OwnerId": "320805250157",
        "IpamPoolId": "ipam-pool-050c886a3ca41cd5b",
        "IpamPoolArn": "arn:aws:ec2::320805250157:ipam-pool/ipam-
pool-050c886a3ca41cd5b",
        "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0a158dde35c51107b",
        "IpamScopeType": "private",
        "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
        "IpamRegion": "us-east-1",
        "Locale": "None",
        "PoolDepth": 1,
        "State": "delete-in-progress",
        "Description": "example",
        "AutoImport": false,
        "AddressFamily": "ipv4",
        "AllocationMinNetmaskLength": 0,
        "AllocationMaxNetmaskLength": 32
    }
}
```

Pour plus d'informations, consultez Supprimer un pool dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DeleteIpamPoolà la section Référence des AWS CLI commandes.

`delete-ipam-resource-discovery`

L'exemple de code suivant montre comment utiliser`delete-ipam-resource-discovery`.

AWS CLI

Pour supprimer une découverte de ressources

Dans cet exemple, vous êtes un administrateur délégué IPAM qui souhaite supprimer une découverte de ressources autre que celle par défaut que vous avez créée pour la partager avec un autre administrateur IPAM lors du processus d'intégration d'IPAM à des comptes extérieurs à votre organisation.

Pour compléter cette demande :

`--region`Il doit s'agir de la région dans laquelle vous avez créé la découverte de ressources. Vous ne pouvez pas supprimer une découverte de ressource par défaut si. `"IsDefault": true` Une découverte de ressources par défaut est une découverte créée automatiquement dans le compte qui crée un IPAM. Pour supprimer une découverte de ressource par défaut, vous devez supprimer l'IPAM.

L'`delete-ipam-resource-discovery`exemple suivant supprime une découverte de ressource.

```
aws ec2 delete-ipam-resource-discovery \
    --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \
    --region us-east-1
```

Sortie :

```
{
    "IpamResourceDiscovery": {
        "OwnerId": "149977607591",
        "IpamResourceDiscoveryId": "ipam-res-disco-0e39761475298ee0f",
        "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0e39761475298ee0f",
        "IpamResourceDiscoveryRegion": "us-east-1",
        "OperatingRegions": [
```

```
            {
                "RegionName": "us-east-1"
            }
        ],
        "IsDefault": false,
        "State": "delete-in-progress"
    }
}
```

Pour plus d'informations sur les découvertes de ressources, consultez la section Travailler avec les découvertes de ressources dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DeleteIpamResourceDiscoveryà la section Référence des AWS CLI commandes.

`delete-ipam-scope`

L'exemple de code suivant montre comment utiliser`delete-ipam-scope`.

AWS CLI

Pour supprimer un scope IPAM

L'`delete-ipam-scope`exemple suivant supprime un IPAM.

```
aws ec2 delete-ipam-scope \
    --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4
```

Sortie :

```
{
    "IpamScope": {
        "OwnerId": "123456789012",
        "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-01c1ebab2b63bd7e4",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "IpamScopeType": "private",
        "IsDefault": false,
        "Description": "Example description",
        "PoolCount": 0,
```

```
        "State": "delete-in-progress"
    }
}
```

Pour plus d'informations, consultez Supprimer une étendue dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DeleteIpamScopeà la section Référence des AWS CLI commandes.

`delete-ipam`

L'exemple de code suivant montre comment utiliser`delete-ipam`.

AWS CLI

Pour supprimer un IPAM

L'`delete-ipam`exemple suivant supprime un IPAM.

```
aws ec2 delete-ipam \
    --ipam-id ipam-036486dfa6af58ee0
```

Sortie :

```
{
    "Ipam": {
        "OwnerId": "123456789012",
        "IpamId": "ipam-036486dfa6af58ee0",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
        "IpamRegion": "us-east-1",
        "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
        "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
        "ScopeCount": 2,
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            },
            {
                "RegionName": "us-east-2"
            },
            {
```

```
                "RegionName": "us-west-1"
            }
        ],
        "State": "delete-in-progress"
    }
}
```

Pour plus d'informations, consultez Supprimer un IPAM dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DeleteIpamà la section Référence des AWS CLI commandes.

`delete-key-pair`

L'exemple de code suivant montre comment utiliser`delete-key-pair`.

AWS CLI

Suppression d'une paire de clés

L'`delete-key-pair`exemple suivant supprime la paire de clés spécifiée.

```
aws ec2 delete-key-pair \
    --key-name my-key-pair
```

Sortie :

```
{
    "Return": true,
    "KeyPairId": "key-03c8d3aceb53b507"
}
```

Pour plus d'informations, voir Création et suppression de paires de clés dans le Guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DeleteKeyPairà la section Référence des AWS CLI commandes.

`delete-launch-template-versions`

L'exemple de code suivant montre comment utiliser`delete-launch-template-versions`.

AWS CLI

Pour supprimer la version d'un modèle de lancement

Cet exemple supprime la version du modèle de lancement spécifiée.

Commande :

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --versions 1
```

Sortie :

```
{
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],
  "SuccessfullyDeletedLaunchTemplateVersions": [
      {
          "LaunchTemplateName": "TestVersion",
          "VersionNumber": 1,
          "LaunchTemplateId": "lt-0abcd290751193123"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteLaunchTemplateVersionsà la section Référence des AWS CLI commandes.

`delete-launch-template`

L'exemple de code suivant montre comment utiliser`delete-launch-template`.

AWS CLI

Suppression d’un modèle de lancement

Cet exemple supprime le modèle de lancement spécifié.

Commande :

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

Sortie :

```
{
  "LaunchTemplate": {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0abcd290751193123",
      "LaunchTemplateName": "TestTemplate",
      "DefaultVersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteLaunchTemplateà la section Référence des AWS CLI commandes.

`delete-local-gateway-route-table-virtual-interface-group-association`

L'exemple de code suivant montre comment utiliser`delete-local-gateway-route-table-virtual-interface-group-association`.

AWS CLI

Pour dissocier une table de routage de passerelle locale d'un groupe d'interfaces virtuelles (VIFs)

L'`delete-local-gateway-route-table-virtual-interface-group-association`exemple suivant supprime l'association entre la table de routage de passerelle locale spécifiée et le groupe VIF.

```
aws ec2 delete-local-gateway-route-table-virtual-interface-group-association \
    --local-gateway-route-table-virtual-interface-group-association-id lgw-vif-grp-assoc-exampleid12345678
```

Sortie :

```
{
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {
        "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-assoc-exampleid12345678",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",
        "LocalGatewayId": "lgw-exampleid11223344",
        "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",
```

```
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-exampleidabcd1234",
        "OwnerId": "111122223333",
        "State": "disassociating",
        "Tags": []
    }
}
```

Pour plus d'informations, consultez les associations de groupes VIF dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociationà la section Référence des AWS CLI commandes.

`delete-local-gateway-route-table-vpc-association`

L'exemple de code suivant montre comment utiliser`delete-local-gateway-route-table-vpc-association`.

AWS CLI

Pour dissocier une table de routage de passerelle locale d'un VPC

L'`delete-local-gateway-route-table-vpc-association`exemple suivant supprime l'association entre la table de routage de passerelle locale spécifiée et le VPC.

```
aws ec2 delete-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-vpc-association-id vpc-example0123456789
```

Sortie :

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-abcd1234wxyz56789",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:555555555555:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
        "LocalGatewayId": "lgw-exampleid01234567",
        "VpcId": "vpc-example0123456789",
        "OwnerId": "555555555555",
        "State": "disassociating"
```

```
    }
}
```

Pour plus d'informations, consultez la section Associations VPC dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DeleteLocalGatewayRouteTableVpcAssociationà la section Référence des AWS CLI commandes.

`delete-local-gateway-route-table`

L'exemple de code suivant montre comment utiliser`delete-local-gateway-route-table`.

AWS CLI

Pour supprimer une table de routage de passerelle locale

L'`delete-local-gateway-route-table`exemple suivant crée une table de routage de passerelle locale avec le mode de routage VPC direct.

```
aws ec2 delete-local-gateway-route-table \
    --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Sortie :

```
{
    "LocalGatewayRouteTable": {
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
        "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
        "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-021345abcdef67890",
        "OwnerId": "111122223333",
        "State": "deleting",
        "Tags": [],
        "Mode": "direct-vpc-routing"
    }
}
```

Pour en savoir plus, consultez Tables de routage de passerelle locale dans le Guide de l'utilisateur AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DeleteLocalGatewayRouteTableà la section Référence des AWS CLI commandes.

`delete-local-gateway-route`

L'exemple de code suivant montre comment utiliser`delete-local-gateway-route`.

AWS CLI

Pour supprimer un itinéraire d'une table de routage de passerelle locale

L'`delete-local-gateway-route`exemple suivant supprime l'itinéraire spécifié de la table de routage de passerelle locale spécifiée.

```
aws ec2 delete-local-gateway-route \
    --destination-cidr-block 0.0.0.0/0 \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

Sortie :

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteLocalGatewayRouteà la section Référence des AWS CLI commandes.

`delete-managed-prefix-list`

L'exemple de code suivant montre comment utiliser`delete-managed-prefix-list`.

AWS CLI

Pour supprimer une liste de préfixes

L'`delete-managed-prefix-list`exemple suivant supprime la liste de préfixes spécifiée.

```
aws ec2 delete-managed-prefix-list \
    --prefix-list-id pl-0123456abcabcabc1
```

Sortie :

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "delete-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "test",
        "MaxEntries": 10,
        "Version": 1,
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DeleteManagedPrefixListà la section Référence des AWS CLI commandes.

`delete-nat-gateway`

L'exemple de code suivant montre comment utiliser`delete-nat-gateway`.

AWS CLI

Pour supprimer une passerelle NAT

Cet exemple supprime la passerelle `nat-04ae55e711cec5680` NAT.

Commande :

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

Sortie :

```
{
    "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteNatGatewayà la section Référence des AWS CLI commandes.

`delete-network-acl-entry`

L'exemple de code suivant montre comment utiliser`delete-network-acl-entry`.

AWS CLI

Pour supprimer une entrée ACL réseau

Cet exemple supprime la règle d'entrée numéro 100 de l'ACL réseau spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkAclEntryà la section Référence des AWS CLI commandes.

`delete-network-acl`

L'exemple de code suivant montre comment utiliser`delete-network-acl`.

AWS CLI

Pour supprimer une ACL réseau

Cet exemple supprime l'ACL réseau spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkAclà la section Référence des AWS CLI commandes.

`delete-network-insights-access-scope-analysis`

L'exemple de code suivant montre comment utiliser`delete-network-insights-access-scope-analysis`.

AWS CLI

Pour supprimer une analyse de l'étendue d'accès au réseau

L'`delete-network-insights-access-scope-analysis`exemple suivant supprime l'analyse de l'étendue d'accès au réseau spécifiée.

```
aws ec2 delete-network-insights-access-scope-analysis \
    --network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

Sortie :

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkInsightsAccessScopeAnalysisà la section Référence des AWS CLI commandes.

`delete-network-insights-access-scope`

L'exemple de code suivant montre comment utiliser`delete-network-insights-access-scope`.

AWS CLI

Pour supprimer une étendue d'accès réseau

L'`delete-network-insights-access-scope`exemple suivant supprime l'étendue d'accès réseau spécifiée.

```
aws ec2 delete-network-insights-access-scope \
    --network-insights-access-scope-id nis-123456789abc01234
```

Sortie :

```
{
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234"
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkInsightsAccessScopeà la section Référence des AWS CLI commandes.

`delete-network-insights-analysis`

L'exemple de code suivant montre comment utiliser`delete-network-insights-analysis`.

AWS CLI

Pour supprimer une analyse de trajectoire

L'`delete-network-insights-analysis`exemple suivant supprime l'analyse spécifiée.

```
aws ec2 delete-network-insights-analysis \
    --network-insights-analysis-id nia-02207aa13eb480c7a
```

Sortie :

```
{
    "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"
}
```

Pour plus d'informations, consultez Getting started using the AWS CLI dans le guide Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkInsightsAnalysisà la section Référence des AWS CLI commandes.

`delete-network-insights-path`

L'exemple de code suivant montre comment utiliser`delete-network-insights-path`.

AWS CLI

Pour supprimer un chemin

L'`delete-network-insights-path`exemple suivant supprime le chemin spécifié. Avant de pouvoir supprimer un chemin, vous devez supprimer toutes ses analyses à l'aide de la `delete-network-insights-analysis` commande.

```
aws ec2 delete-network-insights-path \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

Sortie :

```
{
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"
}
```

Pour plus d'informations, consultez Getting started using the AWS CLI dans le guide Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkInsightsPathà la section Référence des AWS CLI commandes.

`delete-network-interface-permission`

L'exemple de code suivant montre comment utiliser`delete-network-interface-permission`.

AWS CLI

Pour supprimer une autorisation d'interface réseau

Cet exemple supprime l'autorisation d'interface réseau spécifiée.

Commande :

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-perm-06fd19020ede149ea
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkInterfacePermissionà la section Référence des AWS CLI commandes.

`delete-network-interface`

L'exemple de code suivant montre comment utiliser`delete-network-interface`.

AWS CLI

Pour supprimer une interface réseau

Cet exemple supprime l'interface réseau spécifiée. Si la commande réussit, aucune sortie n’est renvoyée.

Commande :

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- Pour plus de détails sur l'API, reportez-vous DeleteNetworkInterfaceà la section Référence des AWS CLI commandes.

`delete-placement-group`

L'exemple de code suivant montre comment utiliser`delete-placement-group`.

AWS CLI

Pour supprimer un groupe de placement

Cet exemple de commande supprime le groupe de placement spécifié.

Commande :

```
aws ec2 delete-placement-group --group-name my-cluster
```

- Pour plus de détails sur l'API, reportez-vous DeletePlacementGroupà la section Référence des AWS CLI commandes.

`delete-queued-reserved-instances`

L'exemple de code suivant montre comment utiliser`delete-queued-reserved-instances`.

AWS CLI

Pour supprimer un achat en file d'attente

L'`delete-queued-reserved-instances`exemple suivant supprime l'instance réservée spécifiée, qui était en attente d'achat.

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

Sortie :

```
{
    "SuccessfulQueuedPurchaseDeletions": [
        {
            "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
        }
    ],
    "FailedQueuedPurchaseDeletions": []
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteQueuedReservedInstancesà la section Référence des AWS CLI commandes.

`delete-route-table`

L'exemple de code suivant montre comment utiliser`delete-route-table`.

AWS CLI

Pour supprimer une table de routage

Cet exemple supprime la table de routage spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- Pour plus de détails sur l'API, reportez-vous DeleteRouteTableà la section Référence des AWS CLI commandes.

`delete-route`

L'exemple de code suivant montre comment utiliser`delete-route`.

AWS CLI

Pour supprimer un itinéraire

Cet exemple supprime l'itinéraire spécifié de la table de routage spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-
block 0.0.0.0/0
```

- Pour plus de détails sur l'API, reportez-vous DeleteRouteà la section Référence des AWS CLI commandes.

`delete-security-group`

L'exemple de code suivant montre comment utiliser`delete-security-group`.

AWS CLI

[EC2-Classique] Pour supprimer un groupe de sécurité

Cet exemple supprime le groupe de sécurité nommé `MySecurityGroup`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] Pour supprimer un groupe de sécurité

Cet exemple supprime le groupe de sécurité dont l'ID est `sg-903004f8`. Notez que vous ne pouvez pas référencer un groupe de sécurité pour EC2 -VPC par son nom. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-security-group --group-id sg-903004f8
```

Pour plus d'informations, veuillez consulter la rubrique Using Security Groups dans le Guide de l'utilisateur AWS Command Line Interface.

- Pour plus de détails sur l'API, reportez-vous DeleteSecurityGroupà la section Référence des AWS CLI commandes.

`delete-snapshot`

L'exemple de code suivant montre comment utiliser`delete-snapshot`.

AWS CLI

Suppression d'un instantané

Cet exemple de commande supprime un instantané dont l'ID de capture est `snap-1234567890abcdef0`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- Pour plus de détails sur l'API, reportez-vous DeleteSnapshotà la section Référence des AWS CLI commandes.

`delete-spot-datafeed-subscription`

L'exemple de code suivant montre comment utiliser`delete-spot-datafeed-subscription`.

AWS CLI

Pour annuler un abonnement au flux de données d'une instance Spot

Cet exemple de commande supprime un abonnement à un flux de données Spot pour le compte. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-spot-datafeed-subscription
```

- Pour plus de détails sur l'API, reportez-vous DeleteSpotDatafeedSubscriptionà la section Référence des AWS CLI commandes.

`delete-subnet-cidr-reservation`

L'exemple de code suivant montre comment utiliser`delete-subnet-cidr-reservation`.

AWS CLI

Pour supprimer une réservation CIDR de sous-réseau

L'`delete-subnet-cidr-reservation`exemple suivant supprime la réservation CIDR de sous-réseau spécifiée.

```
aws ec2 delete-subnet-cidr-reservation \
    --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

Sortie :

```
{
    "DeletedSubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
```

```
    }
}
```

Pour plus d'informations, consultez Subnet CIDR reservations dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DeleteSubnetCidrReservationà la section Référence des AWS CLI commandes.

`delete-subnet`

L'exemple de code suivant montre comment utiliser`delete-subnet`.

AWS CLI

Pour supprimer un sous-réseau

Cet exemple supprime le sous-réseau spécifié. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- Pour plus de détails sur l'API, reportez-vous DeleteSubnetà la section Référence des AWS CLI commandes.

`delete-tags`

L'exemple de code suivant montre comment utiliser`delete-tags`.

AWS CLI

Exemple 1 : pour supprimer un tag d'une ressource

L'`delete-tags`exemple suivant supprime le tag `Stack=Test` de l'image spécifiée. Lorsque vous spécifiez à la fois une valeur et un nom de clé, la balise est supprimée uniquement si la valeur de la balise correspond à la valeur spécifiée.

```
aws ec2 delete-tags \
```

```
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=Test
```

Il est facultatif de spécifier la valeur d'une balise. L'`delete-tags`exemple suivant supprime la balise portant le nom `purpose` de la clé de l'instance spécifiée, quelle que soit la valeur de la balise.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=purpose
```

Si vous spécifiez la chaîne vide comme valeur de balise, la balise est supprimée uniquement si la valeur de la balise est la chaîne vide. L'`delete-tags`exemple suivant indique la chaîne vide comme valeur de balise pour la balise à supprimer.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=Name,Value=
```

Exemple 2 : pour supprimer un tag de plusieurs ressources

L'`delete-tags`exemple suivant supprime le tag``purpose=test`` à la fois d'une instance et d'une AMI. Comme indiqué dans l'exemple précédent, vous pouvez omettre la valeur du tag dans la commande.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 ami-1234567890abcdef0 \
    --tags Key=Purpose
```

- Pour plus de détails sur l'API, reportez-vous DeleteTagsà la section Référence des AWS CLI commandes.

`delete-traffic-mirror-filter-rule`

L'exemple de code suivant montre comment utiliser`delete-traffic-mirror-filter-rule`.

AWS CLI

Pour supprimer une règle de filtrage du trafic miroir

L'`delete-traffic-mirror-filter-rule`exemple suivant supprime la règle de filtrage du trafic spécifiée.

```
aws ec2 delete-traffic-mirror-filter-rule \
    --traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

Sortie :

```
{
    "TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"
}
```

Pour plus d'informations, consultez la section Modifier vos règles de filtrage du trafic dans le guide de mise en miroir AWS du trafic.

- Pour plus de détails sur l'API, reportez-vous DeleteTrafficMirrorFilterRuleà la section Référence des AWS CLI commandes.

`delete-traffic-mirror-filter`

L'exemple de code suivant montre comment utiliser`delete-traffic-mirror-filter`.

AWS CLI

Pour supprimer un filtre antitrafic

L'`delete-traffic-mirror-filter`exemple suivant supprime le filtre de miroir de trafic spécifié.

```
aws ec2 delete-traffic-mirror-filter \
    --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

Sortie :

```
{
    "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"
}
```

Pour plus d'informations, voir Supprimer un filtre de miroir de trafic dans le guide de mise en miroir AWS du trafic.

- Pour plus de détails sur l'API, reportez-vous DeleteTrafficMirrorFilterà la section Référence des AWS CLI commandes.

`delete-traffic-mirror-session`

L'exemple de code suivant montre comment utiliser`delete-traffic-mirror-session`.

AWS CLI

Pour supprimer une session de miroir du trafic

L'`delete-traffic-mirror-session`exemple suivant supprime la session miroir de trafic spécifiée.

```
aws ec2 delete-traffic-mirror-session \
    --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

Sortie :

```
{
    "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"
}
```

Pour plus d'informations, voir Supprimer une session de miroir du trafic dans le guide de mise en miroir AWS du trafic.

- Pour plus de détails sur l'API, reportez-vous DeleteTrafficMirrorSessionà la section Référence des AWS CLI commandes.

`delete-traffic-mirror-target`

L'exemple de code suivant montre comment utiliser`delete-traffic-mirror-target`.

AWS CLI

Pour supprimer une cible miroir du trafic

L'`delete-traffic-mirror-target`exemple suivant supprime la cible de miroir de trafic spécifiée.

```
aws ec2 delete-traffic-mirror-target \
```

```
    --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

Sortie :

```
{
    "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"
}
```

Pour plus d'informations, voir Supprimer une cible de miroir de trafic dans le guide de mise en miroir AWS du trafic.

- Pour plus de détails sur l'API, reportez-vous DeleteTrafficMirrorTargetà la section Référence des AWS CLI commandes.

`delete-transit-gateway-connect-peer`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-connect-peer`.

AWS CLI

Pour supprimer un homologue Transit Gateway Connect

L'`delete-transit-gateway-connect-peer`exemple suivant supprime l'homologue Connect spécifié.

```
aws ec2 delete-transit-gateway-connect-peer \
    --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

Sortie :

```
{
    "TransitGatewayConnectPeer": {
        "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
        "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
        "State": "deleting",
        "CreationTime": "2021-10-13T03:35:17.000Z",
        "ConnectPeerConfiguration": {
            "TransitGatewayAddress": "10.0.0.234",
            "PeerAddress": "172.31.1.11",
            "InsideCidrBlocks": [
                "169.254.6.0/29"
            ],
```

```
            "Protocol": "gre",
            "BgpConfigurations": [
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.2",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                },
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.3",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez les pièces jointes Transit Gateway Connect et les homologues Transit Gateway Connect dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayConnectPeerà la section Référence des AWS CLI commandes.

`delete-transit-gateway-connect`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-connect`.

AWS CLI

Pour supprimer une pièce jointe Connect d'une passerelle de transit

L'`delete-transit-gateway-connect`exemple suivant supprime la pièce jointe Connect spécifiée.

```
aws ec2 delete-transit-gateway-connect \
    --transit-gateway-attachment-id tgw-attach-037012e5dcEXAMPLE
```

Sortie :

```
{
    "TransitGatewayConnect": {
        "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
        "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "deleting",
        "CreationTime": "2021-03-09T19:59:17+00:00",
        "Options": {
            "Protocol": "gre"
        }
    }
}
```

Pour plus d'informations, consultez les pièces jointes Transit Gateway Connect et les homologues Transit Gateway Connect dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayConnectà la section Référence des AWS CLI commandes.

`delete-transit-gateway-multicast-domain`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-multicast-domain`.

AWS CLI

Pour supprimer un domaine de multidiffusion d'une passerelle de transit

L'`delete-transit-gateway-multicast-domain`exemple suivant supprime le domaine de multidiffusion spécifié.

```
aws ec2 delete-transit-gateway-multicast-domain \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Sortie :

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
        "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
        "State": "deleting",
```

```
        "CreationTime": "2019-11-20T22:02:03.000Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des domaines de multidiffusion dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayMulticastDomainà la section Référence des AWS CLI commandes.

`delete-transit-gateway-peering-attachment`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-peering-attachment`.

AWS CLI

Pour supprimer une pièce jointe d'appairage d'une passerelle de transit

L'`delete-transit-gateway-peering-attachment`exemple suivant supprime la pièce jointe d'appairage de passerelle de transit spécifiée.

```
aws ec2 delete-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

Sortie :

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "deleting",
        "CreationTime": "2019-12-09T11:38:31.000Z"
```

```
    }
}
```

Pour plus d'informations, consultez la section Transit Gateway peering Attachments dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayPeeringAttachmentà la section Référence des AWS CLI commandes.

`delete-transit-gateway-policy-table`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-policy-table`.

AWS CLI

Pour supprimer une table de politique de passerelle de transit

L'`delete-transit-gateway-policy-table`exemple suivant supprime la table de politique de passerelle de transit spécifiée.

```
aws ec2 delete-transit-gateway-policy-table \
    --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

Sortie :

```
{
    "TransitGatewayPolicyTables": [
        {
            "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
            "TransitGatewayId": "tgw-067f8505c18f0bd6e",
            "State": "deleting",
            "CreationTime": "2023-11-28T16:36:43+00:00",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez les tableaux des politiques relatives aux passerelles de transit dans le guide de l'utilisateur de Transit Gateway.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayPolicyTableà la section Référence des AWS CLI commandes.

`delete-transit-gateway-prefix-list-reference`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-prefix-list-reference`.

AWS CLI

Pour supprimer une référence à une liste de préfixes

L'`delete-transit-gateway-prefix-list-reference`exemple suivant supprime la référence de liste de préfixes spécifiée.

```
aws ec2 delete-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-1111122222222333
```

Sortie :

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-1111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "deleting",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

Pour plus d'informations, voir les références aux listes de préfixes dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayPrefixListReferenceà la section Référence des AWS CLI commandes.

`delete-transit-gateway-route-table`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-route-table`.

AWS CLI

Pour supprimer une table de routage d'une passerelle de transit

L'`delete-transit-gateway-route-table`exemple suivant supprime la table de routage de la passerelle de transit spécifiée.

```
aws ec2  delete-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE
```

Sortie :

```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "deleting",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-17T20:27:26.000Z"
    }
}
```

Pour plus d'informations, voir Supprimer une table de routage de passerelle de transit dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayRouteTableà la section Référence des AWS CLI commandes.

`delete-transit-gateway-route`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-route`.

AWS CLI

Pour supprimer un bloc CIDR d'une table de routage

L'`delete-transit-gateway-route`exemple suivant supprime le bloc CIDR de la table de routage de la passerelle de transit spécifiée.

```
aws ec2 delete-transit-gateway-route \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \
```

```
    --destination-cidr-block 10.0.2.0/24
```

Sortie :

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-0065acced4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "deleted"
    }
}
```

Pour plus d'informations, voir Supprimer un itinéraire statique dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayRouteà la section Référence des AWS CLI commandes.

`delete-transit-gateway-vpc-attachment`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway-vpc-attachment`.

AWS CLI

Pour supprimer une pièce jointe VPC de passerelle de transit

L'`delete-transit-gateway-vpc-attachment`exemple suivant supprime l'attachement VPC spécifié.

```
aws ec2 delete-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE
```

Sortie :

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "VpcId": "vpc-0065acced4f61c651",
        "VpcOwnerId": "111122223333",
        "State": "deleting",
        "CreationTime": "2019-07-17T16:04:27.000Z"
    }
}
```

Pour plus d'informations, consultez Supprimer une pièce jointe VPC dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayVpcAttachmentà la section Référence des AWS CLI commandes.

`delete-transit-gateway`

L'exemple de code suivant montre comment utiliser`delete-transit-gateway`.

AWS CLI

Pour supprimer une passerelle de transit

L'`delete-transit-gateway`exemple suivant supprime la passerelle de transit spécifiée.

```
aws ec2  delete-transit-gateway \
    --transit-gateway-id tgw-01f04542b2EXAMPLE
```

Sortie :

```
{
    "TransitGateway": {
        "TransitGatewayId": "tgw-01f04542b2EXAMPLE",
        "State": "deleting",
        "OwnerId": "123456789012",
        "Description": "Example Transit Gateway",
        "CreationTime": "2019-08-27T15:04:35.000Z",
        "Options": {
            "AmazonSideAsn": 64515,
```

```
            "AutoAcceptSharedAttachments": "disable",
            "DefaultRouteTableAssociation": "enable",
            "AssociationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",
            "DefaultRouteTablePropagation": "enable",
            "PropagationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",
            "VpnEcmpSupport": "enable",
            "DnsSupport": "enable"
        }
    }
}
```

Pour plus d'informations, voir Supprimer une passerelle de transit dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DeleteTransitGatewayà la section Référence des AWS CLI commandes.

`delete-verified-access-endpoint`

L'exemple de code suivant montre comment utiliser`delete-verified-access-endpoint`.

AWS CLI

Pour supprimer un point de terminaison avec accès vérifié

L'`delete-verified-access-endpoint`exemple suivant supprime le point de terminaison Verified Access spécifié.

```
aws ec2 delete-verified-access-endpoint \
    --verified-access-endpoint-id vae-066fac616d4d546f2
```

Sortie :

```
{
    "VerifiedAccessEndpoint": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
        "ApplicationDomain": "example.com",
        "EndpointType": "network-interface",
        "AttachmentType": "vpc",
```

```
        "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
        "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
        "SecurityGroupIds": [
            "sg-004915970c4c8f13a"
        ],
        "NetworkInterfaceOptions": {
            "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
            "Protocol": "https",
            "Port": 443
        },
        "Status": {
            "Code": "deleting"
        },
        "Description": "Testing Verified Access",
        "CreationTime": "2023-08-25T20:54:43",
        "LastUpdatedTime": "2023-08-25T22:46:32"
    }
}
```

Pour plus d'informations, consultez la section Points de terminaison Verified Access dans le Guide de l'utilisateur AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DeleteVerifiedAccessEndpointà la section Référence des AWS CLI commandes.

`delete-verified-access-group`

L'exemple de code suivant montre comment utiliser`delete-verified-access-group`.

AWS CLI

Pour supprimer un groupe d'accès vérifié

L'`delete-verified-access-group`exemple suivant supprime le groupe d'accès vérifié spécifié.

```
aws ec2 delete-verified-access-group \
    --verified-access-group-id vagr-0dbe967baf14b7235
```

Sortie :

```
{
    "VerifiedAccessGroup": {
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "Owner": "123456789012",
        "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
        "CreationTime": "2023-08-25T19:55:19",
        "LastUpdatedTime": "2023-08-25T22:49:03",
        "DeletionTime": "2023-08-26T00:58:31"
    }
}
```

Pour plus d'informations, consultez la section Groupes d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DeleteVerifiedAccessGroupà la section Référence des AWS CLI commandes.

`delete-verified-access-instance`

L'exemple de code suivant montre comment utiliser`delete-verified-access-instance`.

AWS CLI

Pour supprimer une instance d'accès vérifié

L'`delete-verified-access-instance`exemple suivant supprime l'instance Verified Access spécifiée.

```
aws ec2 delete-verified-access-instance \
    --verified-access-instance-id vai-0ce000c0b7643abea
```

Sortie :

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
```

```
        "LastUpdatedTime": "2023-08-26T01:00:18"
    }
}
```

Pour plus d'informations, consultez la section Instances d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DeleteVerifiedAccessInstanceà la section Référence des AWS CLI commandes.

`delete-verified-access-trust-provider`

L'exemple de code suivant montre comment utiliser`delete-verified-access-trust-provider`.

AWS CLI

Pour supprimer un fournisseur de confiance Verified Access

L'`delete-verified-access-trust-provider`exemple suivant supprime le fournisseur de confiance Verified Access spécifié.

```
aws ec2 delete-verified-access-trust-provider \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Sortie :

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "Testing Verified Access",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T18:40:36",
        "LastUpdatedTime": "2023-08-25T18:40:36"
    }
}
```

Pour plus d'informations, consultez la section Trust providers for Verified Access dans le Guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DeleteVerifiedAccessTrustProviderà la section Référence des AWS CLI commandes.

`delete-volume`

L'exemple de code suivant montre comment utiliser`delete-volume`.

AWS CLI

Pour supprimer un volume

Cet exemple de commande supprime un volume disponible dont l'ID de volume est de`vol-049df61146c4d7901`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- Pour plus de détails sur l'API, reportez-vous DeleteVolumeà la section Référence des AWS CLI commandes.

`delete-vpc-endpoint-connection-notifications`

L'exemple de code suivant montre comment utiliser`delete-vpc-endpoint-connection-notifications`.

AWS CLI

Pour supprimer une notification de connexion d'un terminal

Cet exemple supprime la notification de connexion du point de terminaison spécifiée.

Commande :

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-
ids vpce-nfn-008776de7e03f5abc
```

Sortie :

```
{
  "Unsuccessful": []
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpcEndpointConnectionNotificationsà la section Référence des AWS CLI commandes.

`delete-vpc-endpoint-service-configurations`

L'exemple de code suivant montre comment utiliser`delete-vpc-endpoint-service-configurations`.

AWS CLI

Pour supprimer une configuration de service de point de terminaison

Cet exemple supprime la configuration de service de point de terminaison spécifiée.

Commande :

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

Sortie :

```
{
   "Unsuccessful": []
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpcEndpointServiceConfigurationsà la section Référence des AWS CLI commandes.

`delete-vpc-endpoints`

L'exemple de code suivant montre comment utiliser`delete-vpc-endpoints`.

AWS CLI

Pour supprimer un point de terminaison

Cet exemple supprime les points de terminaison vpce-aa22bb33 et vpce-1a2b3c4d. En cas de réussite partielle ou d'échec de la commande, une liste des éléments ayant échoué est renvoyée. Si la commande aboutit, la liste renvoyée est vide.

Commande :

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

Sortie :

```
{
  "Unsuccessful": []
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpcEndpointsà la section Référence des AWS CLI commandes.

`delete-vpc-peering-connection`

L'exemple de code suivant montre comment utiliser`delete-vpc-peering-connection`.

AWS CLI

Pour supprimer une connexion d'appairage VPC

Cet exemple supprime la connexion d'appairage VPC spécifiée.

Commande :

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Sortie :

```
{
    "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpcPeeringConnectionà la section Référence des AWS CLI commandes.

`delete-vpc`

L'exemple de code suivant montre comment utiliser`delete-vpc`.

AWS CLI

Pour supprimer un VPC

Cet exemple supprime le VPC spécifié. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpcà la section Référence des AWS CLI commandes.

`delete-vpn-connection-route`

L'exemple de code suivant montre comment utiliser`delete-vpn-connection-route`.

AWS CLI

Pour supprimer un itinéraire statique d'une connexion VPN

Cet exemple supprime la route statique spécifiée de la connexion VPN spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-
cidr-block 11.12.0.0/16
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpnConnectionRouteà la section Référence des AWS CLI commandes.

`delete-vpn-connection`

L'exemple de code suivant montre comment utiliser`delete-vpn-connection`.

AWS CLI

Pour supprimer une connexion VPN

Cet exemple supprime la connexion VPN spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpnConnectionà la section Référence des AWS CLI commandes.

`delete-vpn-gateway`

L'exemple de code suivant montre comment utiliser`delete-vpn-gateway`.

AWS CLI

Pour supprimer une passerelle privée virtuelle

Cet exemple supprime la passerelle privée virtuelle spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- Pour plus de détails sur l'API, reportez-vous DeleteVpnGatewayà la section Référence des AWS CLI commandes.

`deprovision-byoip-cidr`

L'exemple de code suivant montre comment utiliser`deprovision-byoip-cidr`.

AWS CLI

Pour empêcher l'utilisation d'une plage d'adresses IP

L'exemple suivant supprime l'utilisation de la plage d'adresses spécifiée AWS.

```
aws ec2 deprovision-byoip-cidr \
    --cidr 203.0.113.25/24
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-deprovision"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeprovisionByoipCidrà la section Référence des AWS CLI commandes.

`deprovision-ipam-pool-cidr`

L'exemple de code suivant montre comment utiliser`deprovision-ipam-pool-cidr`.

AWS CLI

Pour déprovisionner un pool IPAM CIDR

L'`deprovision-ipam-pool-cidr`exemple suivant déprovisionne un CIDR fourni à un pool IPAM.

(Linux) :

```
aws ec2 deprovision-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-02ec043a19bbe5d08 \
    --cidr 11.0.0.0/16
```

(Fenêtres) :

```
aws ec2 deprovision-ipam-pool-cidr ^
    --ipam-pool-id ipam-pool-02ec043a19bbe5d08 ^
    --cidr 11.0.0.0/16
```

Sortie :

```
{
    "IpamPoolCidr": {
        "Cidr": "11.0.0.0/16",
```

```
        "State": "pending-deprovision"
    }
}
```

Pour plus d'informations, consultez le pool de déprovisionnement CIDRs dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DeprovisionIpamPoolCidrà la section Référence des AWS CLI commandes.

`deregister-image`

L'exemple de code suivant montre comment utiliser`deregister-image`.

AWS CLI

Pour annuler l'enregistrement d'un AMI

Cet exemple désenregistre l'AMI spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- Pour plus de détails sur l'API, reportez-vous DeregisterImageà la section Référence des AWS CLI commandes.

`deregister-instance-event-notification-attributes`

L'exemple de code suivant montre comment utiliser`deregister-instance-event-notification-attributes`.

AWS CLI

Exemple 1 : pour supprimer toutes les balises des notifications d'événements

L'`deregister-instance-event-notification-attributes`exemple suivant supprime`IncludeAllTagsOfInstance=true`, ce qui a pour effet de `IncludeAllTagsOfInstance` définir sur`false`.

```
aws ec2 deregister-instance-event-notification-attributes \
```

```
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

Sortie :

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

Pour plus d'informations, consultez la section Événements planifiés pour vos instances dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

Exemple 2 : pour supprimer des balises spécifiques des notifications d'événements

L'`deregister-instance-event-notification-attributes`exemple suivant supprime la balise spécifiée des balises incluses dans les notifications d'événements. Pour décrire les autres balises incluses dans les notifications d'événements, utilisez`describe-instance-event-notification-attributes`.

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute InstanceTagKeys="tag-key2"
```

Sortie :

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [
            "tag-key2"
        ],
        "IncludeAllTagsOfInstance": false
    }
}
```

Pour plus d'informations, consultez la section Événements planifiés pour vos instances dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DeregisterInstanceEventNotificationAttributesà la section Référence des AWS CLI commandes.

`deregister-transit-gateway-multicast-group-members`

L'exemple de code suivant montre comment utiliser`deregister-transit-gateway-multicast-group-members`.

AWS CLI

Pour désenregistrer les membres d'un groupe de multidiffusion

Cet exemple désenregistre le membre du groupe d'interface réseau spécifié du groupe de multidiffusion de la passerelle de transit.

```
aws ec2 deregister-transit-gateway-multicast-group-members \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-0e246d3269EXAMPLE
```

Sortie :

```
{
    "DeregisteredMulticastGroupMembers": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
        "RegisteredNetworkInterfaceIds": [
            "eni-0e246d3269EXAMPLE"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

Pour plus d'informations, consultez la section Désenregistrer les membres d'un groupe de multidiffusion dans le guide de l'utilisateur de AWS Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DeregisterTransitGatewayMulticastGroupMembersà la section Référence des AWS CLI commandes.

`deregister-transit-gateway-multicast-group-source`

L'exemple de code suivant montre comment utiliser`deregister-transit-gateway-multicast-group-source`.

AWS CLI

Pour désenregistrer une source du groupe de multidiffusion de la passerelle de transit

Cet exemple désenregistre la source du groupe d'interfaces réseau spécifiée du groupe de multidiffusion.

```
aws ec2 register-transit-gateway-multicast-group-sources \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-07f290fc3c090cbae
```

Sortie :

```
{
    "DeregisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "DeregisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

Pour plus d'informations, consultez la section Désenregistrer les sources d'un groupe de multidiffusion dans le guide de l'utilisateur de AWS Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DeregisterTransitGatewayMulticastGroupSourceà la section Référence des AWS CLI commandes.

`describe-account-attributes`

L'exemple de code suivant montre comment utiliser`describe-account-attributes`.

AWS CLI

Pour décrire tous les attributs de votre AWS compte

Cet exemple décrit les attributs de votre AWS compte.

Commande :

```
aws ec2 describe-account-attributes
```

Sortie :

```
{
    "AccountAttributes": [
        {
            "AttributeName": "vpc-max-security-groups-per-interface",
            "AttributeValues": [
                {
                    "AttributeValue": "5"
                }
            ]
        },
        {
            "AttributeName": "max-instances",
            "AttributeValues": [
                {
                    "AttributeValue": "20"
                }
            ]
        },
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        },
        {
            "AttributeName": "default-vpc",
            "AttributeValues": [
                {
                    "AttributeValue": "none"
                }
            ]
        },
        {
            "AttributeName": "max-elastic-ips",
```

```
            "AttributeValues": [
                {
                    "AttributeValue": "5"
                }
            ]
        },
        {
            "AttributeName": "vpc-max-elastic-ips",
            "AttributeValues": [
                {
                    "AttributeValue": "5"
                }
            ]
        }
    ]
}
```

Pour décrire un attribut unique de votre AWS compte

Cet exemple décrit l'`supported-platforms`attribut de votre AWS compte.

Commande :

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

Sortie :

```
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAccountAttributesà la section Référence des AWS CLI commandes.

`describe-address-transfers`

L'exemple de code suivant montre comment utiliser`describe-address-transfers`.

AWS CLI

Pour décrire un transfert d'adresse IP élastique

L'`describe-address-transfers`exemple suivant décrit le transfert d'adresse IP élastique pour l'adresse IP élastique spécifiée.

```
aws ec2 describe-address-transfers \
    --allocation-ids eipalloc-09ad461b0d03f6aaf
```

Sortie :

```
{
    "AddressTransfers": [
        {
            "PublicIp": "100.21.184.216",
            "AllocationId": "eipalloc-09ad461b0d03f6aaf",
            "TransferAccountId": "123456789012",
            "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",
            "AddressTransferStatus": "pending"
        }
    ]
}
```

Pour plus d'informations, consultez Transfer Elastic IP address dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeAddressTransfersà la section Référence des AWS CLI commandes.

`describe-addresses-attribute`

L'exemple de code suivant montre comment utiliser`describe-addresses-attribute`.

AWS CLI

Pour afficher les attributs du nom de domaine associés à une adresse IP élastique

Les `describe-addresses-attribute` exemples suivants renvoient les attributs du nom de domaine associé à l'adresse IP élastique.

Linux :

```
aws ec2 describe-addresses-attribute \
    --allocation-ids eipalloc-abcdef01234567890 \
    --attribute domain-name
```

Windows :

```
aws ec2 describe-addresses-attribute ^
    --allocation-ids eipalloc-abcdef01234567890 ^
    --attribute domain-name
```

Sortie :

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com."
        }
    ]
}
```

Pour afficher les attributs d'une adresse IP élastique, vous devez d'abord avoir associé un nom de domaine à l'adresse IP élastique. Pour plus d'informations, consultez la section Utiliser le DNS inversé pour les applications de messagerie dans le guide de EC2 l'utilisateur Amazon ou modify-address-attributedans le manuel de référence des commandes AWS CLI.

- Pour plus de détails sur l'API, reportez-vous DescribeAddressesAttributeà la section Référence des AWS CLI commandes.

`describe-addresses`

L'exemple de code suivant montre comment utiliser`describe-addresses`.

AWS CLI

Exemple 1 : récupération des détails de toutes vos adresses IP Elastic

L'exemple `describe addresses` suivant affiche les détails de vos adresses IP Elastic.

```
aws ec2 describe-addresses
```

Sortie :

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "198.51.100.0",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        },
        {
            "Domain": "vpc",
            "PublicIpv4Pool": "amazon",
            "InstanceId": "i-1234567890abcdef0",
            "NetworkInterfaceId": "eni-12345678",
            "AssociationId": "eipassoc-12345678",
            "NetworkInterfaceOwnerId": "123456789012",
            "PublicIp": "203.0.113.0",
            "AllocationId": "eipalloc-12345678",
            "PrivateIpAddress": "10.0.1.241"
        }
    ]
}
```

Exemple 2 : Pour récupérer les détails de vos adresses IP Elastic pour EC2 -VPC

L'exemple `describe-addresses` suivant affiche les détails de vos adresses IP Elastic à utiliser avec des instances dans un VPC.

```
aws ec2 describe-addresses \
    --filters "Name=domain,Values=vpc"
```

Sortie :

```
{
    "Addresses": [
        {
            "Domain": "vpc",
            "PublicIpv4Pool": "amazon",
            "InstanceId": "i-1234567890abcdef0",
            "NetworkInterfaceId": "eni-12345678",
            "AssociationId": "eipassoc-12345678",
            "NetworkInterfaceOwnerId": "123456789012",
            "PublicIp": "203.0.113.0",
            "AllocationId": "eipalloc-12345678",
            "PrivateIpAddress": "10.0.1.241"
        }
    ]
}
```

Exemple 3 : récupération des détails d'une adresse IP Elastic spécifiée selon l'identifiant d'allocation

L'`describe-addresses`exemple suivant affiche des détails sur l'adresse IP élastique avec l'ID d'allocation spécifié, qui est associée à une instance dans EC2 -VPC.

```
aws ec2 describe-addresses \
    --allocation-ids eipalloc-282d9641
```

Sortie :

```
{
    "Addresses": [
        {
            "Domain": "vpc",
            "PublicIpv4Pool": "amazon",
            "InstanceId": "i-1234567890abcdef0",
            "NetworkInterfaceId": "eni-1a2b3c4d",
            "AssociationId": "eipassoc-123abc12",
            "NetworkInterfaceOwnerId": "1234567891012",
            "PublicIp": "203.0.113.25",
            "AllocationId": "eipalloc-282d9641",
            "PrivateIpAddress": "10.251.50.12"
        }
    ]
}
```

Exemple 4 : récupération des détails d'une adresse IP Elastic spécifiée au moyen de son adresse IP privée de VPC

L'`describe-addresses`exemple suivant affiche des détails sur l'adresse IP élastique associée à une adresse IP privée particulière dans EC2 -VPC.

```
aws ec2 describe-addresses \
    --filters "Name=private-ip-address,Values=10.251.50.12"
```

Exemple 5 : Pour récupérer des informations sur les adresses IP Elastic dans EC2 -Classic

TThe L'`describe-addresses`exemple suivant affiche les détails de vos adresses IP élastiques à utiliser dans EC2 -Classic.

```
aws ec2 describe-addresses \
    --filters "Name=domain,Values=standard"
```

Sortie :

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "203.0.110.25",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        }
    ]
}
```

Exemple 6 : récupération des détails d'une adresse IP Elastic spécifiée au moyen de son adresse IP publique

L'`describe-addresses`exemple suivant affiche des détails sur l'adresse IP élastique avec la valeur`203.0.110.25`, qui est associée à une instance dans EC2 -Classic.

```
aws ec2 describe-addresses \
    --public-ips 203.0.110.25
```

Sortie :

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "203.0.110.25",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAddressesà la section Référence des AWS CLI commandes.

`describe-aggregate-id-format`

L'exemple de code suivant montre comment utiliser`describe-aggregate-id-format`.

AWS CLI

Pour décrire les paramètres de format d'identification plus longs pour tous les types de ressources d'une région

L'`describe-aggregate-id-format`exemple suivant décrit l'état général du format d'identifiant long pour la région actuelle. La `Deadline` valeur indique que les délais pour que ces ressources passent définitivement du format d'identifiant court au format d'identifiant long ont expiré. La `UseLongIdsAggregated` valeur indique que tous les utilisateurs et rôles IAM sont configurés pour utiliser le format d'identifiant long pour tous les types de ressources.

```
aws ec2 describe-aggregate-id-format
```

Sortie :

```
{
    "UseLongIdsAggregated": true,
    "Statuses": [
        {
            "Deadline": "2018-08-13T02:00:00.000Z",
            "Resource": "network-interface-attachment",
            "UseLongIds": true
        },
```

```
        {
            "Deadline": "2016-12-13T02:00:00.000Z",
            "Resource": "instance",
            "UseLongIds": true
        },
        {
            "Deadline": "2018-08-13T02:00:00.000Z",
            "Resource": "elastic-ip-association",
            "UseLongIds": true
        },
        ...
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAggregateIdFormatà la section Référence des AWS CLI commandes.

`describe-availability-zones`

L'exemple de code suivant montre comment utiliser`describe-availability-zones`.

AWS CLI

Description de vos zones de disponibilité

L'exemple suivant `describe-availability-zones` affiche les détails des zones de disponibilité mises à votre disposition. La réponse n'inclut que les zones de disponibilité pour la région actuelle. Dans cet exemple, la région par défaut du profil `us-west-2` (Oregon) est utilisée.

```
aws ec2 describe-availability-zones
```

Sortie :

```
{
    "AvailabilityZones": [
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2a",
            "ZoneId": "usw2-az1",
```

```
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2b",
            "ZoneId": "usw2-az2",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2c",
            "ZoneId": "usw2-az3",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2d",
            "ZoneId": "usw2-az4",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opted-in",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2-lax-1a",
            "ZoneId": "usw2-lax1-az1",
            "GroupName": "us-west-2-lax-1",
            "NetworkBorderGroup": "us-west-2-lax-1"
        }
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAvailabilityZonesà la section Référence des AWS CLI commandes.

`describe-aws-network-performance-metric-subscription`

L'exemple de code suivant montre comment utiliser`describe-aws-network-performance-metric-subscription`.

AWS CLI

Pour décrire vos abonnements aux métriques

L'`describe-aws-network-performance-metric-subscriptions`exemple suivant décrit vos abonnements aux métriques.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Sortie :

```
{
    "Subscriptions": [
        {
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les abonnements dans le Guide de l'utilisateur sur les performances de l'infrastructure.

- Pour plus de détails sur l'API, reportez-vous DescribeAwsNetworkPerformanceMetricSubscriptionà la section Référence des AWS CLI commandes.

`describe-aws-network-performance-metric-subscriptions`

L'exemple de code suivant montre comment utiliser`describe-aws-network-performance-metric-subscriptions`.

AWS CLI

Pour décrire vos abonnements aux métriques

L'`describe-aws-network-performance-metric-subscriptions`exemple suivant décrit vos abonnements aux métriques.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Sortie :

```
{
    "Subscriptions": [
        {
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les abonnements dans le Guide de l'utilisateur sur les performances de l'infrastructure.

- Pour plus de détails sur l'API, reportez-vous DescribeAwsNetworkPerformanceMetricSubscriptionsà la section Référence des AWS CLI commandes.

`describe-bundle-tasks`

L'exemple de code suivant montre comment utiliser`describe-bundle-tasks`.

AWS CLI

Pour décrire les tâches de votre bundle

Cet exemple décrit toutes vos tâches groupées.

Commande :

```
aws ec2 describe-bundle-tasks
```

Sortie :

```
{
  "BundleTasks": [
    {
      "UpdateTime": "2015-09-15T13:26:54.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "Storage": {
        "S3": {
            "Prefix": "winami",
            "Bucket": "bundletasks"
        }
      },
      "State": "bundling",
      "StartTime": "2015-09-15T13:24:35.000Z",
      "Progress": "3%",
      "BundleId": "bun-2a4e041c"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeBundleTasksà la section Référence des AWS CLI commandes.

`describe-byoip-cidrs`

L'exemple de code suivant montre comment utiliser`describe-byoip-cidrs`.

AWS CLI

Pour décrire vos plages d'adresses provisionnées

L'`describe-byoip-cidrs`exemple suivant affiche des détails sur les plages d' IPv4 adresses publiques que vous avez configurées pour être utilisées par AWS.

```
aws ec2 describe-byoip-cidrs
```

Sortie :

```
{
    "ByoipCidrs": [
        {
            "Cidr": "203.0.113.25/24",
            "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
            "State": "provisioned"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeByoipCidrsà la section Référence des AWS CLI commandes.

`describe-capacity-reservation-fleets`

L'exemple de code suivant montre comment utiliser`describe-capacity-reservation-fleets`.

AWS CLI

Pour consulter une flotte de réservation de capacité

L'`describe-capacity-reservation-fleets`exemple suivant répertorie les informations de configuration et de capacité pour le parc de réservation de capacité spécifié. Il contient également des détails sur les réservations de capacité individuelles qui se trouvent au sein de la flotte.

```
aws ec2 describe-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

Sortie :

```
{
    "CapacityReservationFleets": [
        {
            "State": "active",
            "EndDate": "2022-12-31T23:59:59.000Z",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "CapacityReservationFleetId": "crf-abcdef01234567890",
            "Tenancy": "default",
            "InstanceTypeSpecifications": [
```

```
                {
                    "CapacityReservationId": "cr-1234567890abcdef0",
                    "AvailabilityZone": "us-east-1a",
                    "FulfilledCapacity": 5.0,
                    "Weight": 1.0,
                    "CreateDate": "2022-07-02T08:34:33.398Z",
                    "InstancePlatform": "Linux/UNIX",
                    "TotalInstanceCount": 5,
                    "Priority": 1,
                    "EbsOptimized": true,
                    "InstanceType": "m5.xlarge"
                }
            ],
            "TotalTargetCapacity": 5,
            "TotalFulfilledCapacity": 5.0,
            "CreateTime": "2022-07-02T08:34:33.397Z",
            "AllocationStrategy": "prioritized"
        }
    ]
}
```

Pour plus d'informations sur les flottes de réservation de capacité, consultez la section Flottes de réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeCapacityReservationFleetsà la section Référence des AWS CLI commandes.

`describe-capacity-reservations`

L'exemple de code suivant montre comment utiliser`describe-capacity-reservations`.

AWS CLI

Exemple 1 : Pour décrire une ou plusieurs de vos réservations de capacité

L'`describe-capacity-reservations`exemple suivant affiche les détails de toutes vos réservations de capacité dans la AWS région actuelle.

```
aws ec2 describe-capacity-reservations
```

Sortie :

```
{
```

```
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-1234abcd56EXAMPLE",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c5.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "active",
            "StartDate": "2024-10-23T15:00:24+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:00:24+00:00",
            "Tags": [],
            "CapacityAllocations": []
        },
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-abcdEXAMPLE9876ef",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c4.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "cancelled",
            "StartDate": "2024-10-23T15:01:03+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:01:02+00:00",
            "Tags": [],
            "CapacityAllocations": []
```

```
        }
    ]
}
```

Exemple 2 : Pour décrire une ou plusieurs de vos réservations de capacité

L'`describe-capacity-reservations`exemple suivant affiche les détails relatifs à la réservation de capacité spécifiée.

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-ids cr-1234abcd56EXAMPLE
```

Sortie :

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-abcdEXAMPLE9876ef",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c4.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "active",
            "StartDate": "2024-10-23T15:01:03+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:01:02+00:00",
            "Tags": [],
            "CapacityAllocations": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher une réservation de capacité dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeCapacityReservationsà la section Référence des AWS CLI commandes.

`describe-carrier-gateways`

L'exemple de code suivant montre comment utiliser`describe-carrier-gateways`.

AWS CLI

Pour décrire toutes les passerelles des opérateurs

L'`describe-carrier-gateways`exemple suivant répertorie toutes les passerelles de votre opérateur.

```
aws ec2 describe-carrier-gateways
```

Sortie :

```
{
    "CarrierGateways": [
        {
            "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
            "VpcId": "vpc-0c529aEXAMPLE",
            "State": "available",
            "OwnerId": "123456789012",
            "Tags": [
                {

                    "Key": "example",
                    "Value": "tag"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Carrier gateways< https://docs.aws.amazon.com/vpc/ latest/ userguide/Carrier _Gateway.html> dans le guide de l'utilisateur d'Amazon Virtual Private Cloud.

- Pour plus de détails sur l'API, reportez-vous DescribeCarrierGatewaysà la section Référence des AWS CLI commandes.

`describe-classic-link-instances`

L'exemple de code suivant montre comment utiliser`describe-classic-link-instances`.

AWS CLI

Pour décrire les instances EC2 -Classic liées

Cet exemple répertorie toutes vos instances EC2 -Classic liées.

Commande :

```
aws ec2 describe-classic-link-instances
```

Sortie :

```
{
        "Instances": [
                {
                        "InstanceId": "i-1234567890abcdef0",
                        "VpcId": "vpc-88888888",
                        "Groups": [
                                {
                                        "GroupId": "sg-11122233"
                                }
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance",
                                        "Key": "Name"
                                }
                        ]
                },
                {
                        "InstanceId": "i-0598c7d356eba48d7",
                        "VpcId": "vpc-12312312",
                        "Groups": [
                                {
                                        "GroupId": "sg-aabbccdd"
                                }
```

```
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance2",
                                        "Key": "Name"
                                }
                        ]
                }
        ]
}
```

Cet exemple répertorie toutes vos instances EC2 -Classic liées et filtre la réponse pour n'inclure que les instances liées au VPC vpc-88888888.

Commande :

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

Sortie :

```
{
        "Instances": [
                {
                        "InstanceId": "i-1234567890abcdef0",
                        "VpcId": "vpc-88888888",
                        "Groups": [
                                {
                                        "GroupId": "sg-11122233"
                                }
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance",
                                        "Key": "Name"
                                }
                        ]
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeClassicLinkInstancesà la section Référence des AWS CLI commandes.

`describe-client-vpn-authorization-rules`

L'exemple de code suivant montre comment utiliser`describe-client-vpn-authorization-rules`.

AWS CLI

Pour décrire les règles d'autorisation pour un point de terminaison VPN client

L'`describe-client-vpn-authorization-rules`exemple suivant affiche des détails sur les règles d'autorisation pour le point de terminaison VPN du Client spécifié.

```
aws ec2 describe-client-vpn-authorization-rules \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "AuthorizationRules": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "GroupId": "",
            "AccessAll": true,
            "DestinationCidr": "0.0.0.0/0",
            "Status": {
                "Code": "active"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Règles d'autorisation dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DescribeClientVpnAuthorizationRulesà la section Référence des AWS CLI commandes.

`describe-client-vpn-connections`

L'exemple de code suivant montre comment utiliser`describe-client-vpn-connections`.

AWS CLI

Pour décrire les connexions à un point de terminaison VPN client

L'`describe-client-vpn-connections`exemple suivant affiche des détails sur les connexions client au point de terminaison VPN client spécifié.

```
aws ec2 describe-client-vpn-connections \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "Connections": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Timestamp": "2019-08-12 07:58:34",
            "ConnectionId": "cvpn-connection-0e03eb24267165acd",
            "ConnectionEstablishedTime": "2019-08-12 07:57:14",
            "IngressBytes": "32302",
            "EgressBytes": "5696",
            "IngressPackets": "332",
            "EgressPackets": "67",
            "ClientIp": "172.31.0.225",
            "CommonName": "client1.domain.tld",
            "Status": {
                "Code": "terminated"
            },
            "ConnectionEndTime": "2019-08-12 07:58:34"
        },
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Timestamp": "2019-08-12 08:02:54",
            "ConnectionId": "cvpn-connection-00668867a40f18253",
            "ConnectionEstablishedTime": "2019-08-12 08:02:53",
            "IngressBytes": "2951",
            "EgressBytes": "2611",
            "IngressPackets": "9",
            "EgressPackets": "6",
            "ClientIp": "172.31.0.226",
            "CommonName": "client1.domain.tld",
            "Status": {
                "Code": "active"
```

```
            },
            "ConnectionEndTime": "-"
        }
    ]
}
```

Pour plus d'informations, consultez la section Connexions client dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DescribeClientVpnConnectionsà la section Référence des AWS CLI commandes.

`describe-client-vpn-endpoints`

L'exemple de code suivant montre comment utiliser`describe-client-vpn-endpoints`.

AWS CLI

Pour décrire les points de terminaison VPN de votre Client

L'`describe-client-vpn-endpoints`exemple suivant affiche les détails de tous les points de terminaison VPN de votre Client.

```
aws ec2 describe-client-vpn-endpoints
```

Sortie :

```
{
    "ClientVpnEndpoints": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Description": "Endpoint for Admin access",
            "Status": {
                "Code": "available"
            },
            "CreationTime": "2020-11-13T11:37:27",
            "DnsName": "*.cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com",
            "ClientCidrBlock": "172.31.0.0/16",
            "DnsServers": [
                "8.8.8.8"
            ],
            "SplitTunnel": false,
```

```
            "VpnProtocol": "openvpn",
            "TransportProtocol": "udp",
            "VpnPort": 443,
            "ServerCertificateArn": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "AuthenticationOptions": [
                {
                    "Type": "certificate-authentication",
                    "MutualAuthentication": {
                        "ClientRootCertificateChain": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
                    }
                }
            ],
            "ConnectionLogOptions": {
                "Enabled": true,
                "CloudwatchLogGroup": "Client-vpn-connection-logs",
                "CloudwatchLogStream": "cvpn-endpoint-123456789123abcde-ap-
south-1-2020/11/13-FCD8HEMVaCcw"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Client VPN"
                }
            ],
            "SecurityGroupIds": [
                "sg-aabbcc11223344567"
            ],
            "VpcId": "vpc-a87f92c1",
            "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/
endpoints/cvpn-endpoint-123456789123abcde",
            "ClientConnectOptions": {
                 "Enabled": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Points de terminaison du VPN client dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DescribeClientVpnEndpointsà la section Référence des AWS CLI commandes.

`describe-client-vpn-routes`

L'exemple de code suivant montre comment utiliser`describe-client-vpn-routes`.

AWS CLI

Pour décrire les itinéraires d'un point de terminaison VPN client

L'`describe-client-vpn-routes`exemple suivant affiche des détails sur les itinéraires pour le point de terminaison VPN du Client spécifié.

```
aws ec2 describe-client-vpn-routes \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "Routes": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "DestinationCidr": "10.0.0.0/16",
            "TargetSubnet": "subnet-0123456789abcabca",
            "Type": "Nat",
            "Origin": "associate",
            "Status": {
                "Code": "active"
            },
            "Description": "Default Route"
        },
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "DestinationCidr": "0.0.0.0/0",
            "TargetSubnet": "subnet-0123456789abcabca",
            "Type": "Nat",
            "Origin": "add-route",
            "Status": {
                "Code": "active"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Routes dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DescribeClientVpnRoutesà la section Référence des AWS CLI commandes.

`describe-client-vpn-target-networks`

L'exemple de code suivant montre comment utiliser`describe-client-vpn-target-networks`.

AWS CLI

Pour décrire les réseaux cibles d'un point de terminaison VPN client

L'`describe-client-vpn-target-networks`exemple suivant affiche des détails sur les réseaux cibles pour le point de terminaison VPN du Client spécifié.

```
aws ec2 describe-client-vpn-target-networks \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "ClientVpnTargetNetworks": [
        {
            "AssociationId": "cvpn-assoc-012e837060753dc3d",
            "VpcId": "vpc-11111222222333333",
            "TargetNetworkId": "subnet-0123456789abcabca",
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Status": {
                "Code": "associating"
            },
            "SecurityGroups": [
                "sg-012345678910abcab"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Réseaux cibles dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DescribeClientVpnTargetNetworksà la section Référence des AWS CLI commandes.

`describe-coip-pools`

L'exemple de code suivant montre comment utiliser`describe-coip-pools`.

AWS CLI

Pour décrire les pools d'adresses IP appartenant au client

L'`describe-coip-pools`exemple suivant décrit les pools d'adresses IP appartenant au client dans votre AWS compte.

```
aws ec2 describe-coip-pools
```

Sortie :

```
{
    "CoipPools": [
        {
            "PoolId": "ipv4pool-coip-123a45678bEXAMPLE",
            "PoolCidrs": [
                "0.0.0.0/0"
            ],
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
            "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-123a45678bEXAMPLE"
        }
    ]
}
```

Pour plus d'informations, voir Adresses IP appartenant au client dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DescribeCoipPoolsà la section Référence des AWS CLI commandes.

`describe-conversion-tasks`

L'exemple de code suivant montre comment utiliser`describe-conversion-tasks`.

AWS CLI

Pour consulter le statut d'une tâche de conversion

Cet exemple renvoie le statut d'une tâche de conversion portant l'ID import-i-ffvko 9js.

Commande :

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

Sortie :

```
{
    "ConversionTasks": [
        {
            "ConversionTaskId": "import-i-ffvko9js",
            "ImportInstance": {
                "InstanceId": "i-1234567890abcdef0",
                "Volumes": [
                    {
                        "Volume": {
                            "Id": "vol-049df61146c4d7901",
                            "Size": 16
                        },
                        "Status": "completed",
                        "Image": {
                            "Size": 1300687360,
                            "ImportManifestUrl": "https://s3.amazonaws.com/
myimportbucket/411443cd-d620-4f1c-9d66-13144EXAMPLE/RHEL5.vmdkmanifest.xml?
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=140EXAMPLE&Signature=XYNhznHNgCqsjDxL9wRL
%2FJvEXAMPLE",
                            "Format": "VMDK"
                        },
                        "BytesConverted": 1300682960,
                        "AvailabilityZone": "us-east-1d"
                    }
                ]
            },
            "ExpirationTime": "2014-05-14T22:06:23Z",
            "State": "completed"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeConversionTasksà la section Référence des AWS CLI commandes.

`describe-customer-gateways`

L'exemple de code suivant montre comment utiliser`describe-customer-gateways`.

AWS CLI

Pour décrire les passerelles de vos clients

Cet exemple décrit vos passerelles client.

Commande :

```
aws ec2 describe-customer-gateways
```

Sortie :

```
{
    "CustomerGateways": [
        {
            "CustomerGatewayId": "cgw-b4dc3961",
            "IpAddress": "203.0.113.12",
            "State": "available",
            "Type": "ipsec.1",
            "BgpAsn": "65000"
        },
        {
            "CustomerGatewayId": "cgw-0e11f167",
            "IpAddress": "12.1.2.3",
            "State": "available",
            "Type": "ipsec.1",
            "BgpAsn": "65534"
        }
    ]
}
```

Pour décrire une passerelle client spécifique

Cet exemple décrit la passerelle client spécifiée.

Commande :

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

Sortie :

```
{
    "CustomerGateways": [
        {
            "CustomerGatewayId": "cgw-0e11f167",
            "IpAddress": "12.1.2.3",
            "State": "available",
            "Type": "ipsec.1",
            "BgpAsn": "65534"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeCustomerGatewaysà la section Référence des AWS CLI commandes.

`describe-dhcp-options`

L'exemple de code suivant montre comment utiliser`describe-dhcp-options`.

AWS CLI

Exemple 1 : Pour décrire vos options DHCP

L'`describe-dhcp-options`exemple suivant permet de récupérer des informations sur vos options DHCP.

```
aws ec2 describe-dhcp-options
```

Sortie :

```
{
    "DhcpOptions": [
        {
            "DhcpConfigurations": [
                {
                    "Key": "domain-name",
                    "Values": [
                        {
                            "Value": "us-east-2.compute.internal"
                        }
```

```
                    ]
                },
                {
                    "Key": "domain-name-servers",
                    "Values": [
                        {
                            "Value": "AmazonProvidedDNS"
                        }
                    ]
                }
            ],
            "DhcpOptionsId": "dopt-19edf471",
            "OwnerId": "111122223333"
        },
        {
            "DhcpConfigurations": [
                {
                    "Key": "domain-name",
                    "Values": [
                        {
                            "Value": "us-east-2.compute.internal"
                        }
                    ]
                },
                {
                    "Key": "domain-name-servers",
                    "Values": [
                        {
                            "Value": "AmazonProvidedDNS"
                        }
                    ]
                }
            ],
            "DhcpOptionsId": "dopt-fEXAMPLE",
            "OwnerId": "111122223333"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des ensembles d'options DHCP dans le guide de l'AWS utilisateur VPC.

Exemple 2 : pour décrire vos options DHCP et filtrer la sortie

L'`describe-dhcp-options`exemple suivant décrit vos options DHCP et utilise un filtre pour renvoyer uniquement les options DHCP correspondant au serveur `example.com` de noms de domaine. L'exemple utilise le `--query` paramètre pour afficher uniquement les informations de configuration et l'ID dans la sortie.

```
aws ec2 describe-dhcp-options \
    --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
    --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"
```

Sortie :

```
[
    [
        [
            {
                "Key": "domain-name",
                "Values": [
                    {
                        "Value": "example.com"
                    }
                ]
            },
            {
                "Key": "domain-name-servers",
                "Values": [
                    {
                        "Value": "172.16.16.16"
                    }
                ]
            }
        ],
        "dopt-001122334455667ab"
    ]
]
```

Pour plus d'informations, consultez la section Utilisation des ensembles d'options DHCP dans le guide de l'AWS utilisateur VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeDhcpOptionsà la section Référence des AWS CLI commandes.

`describe-egress-only-internet-gateways`

L'exemple de code suivant montre comment utiliser`describe-egress-only-internet-gateways`.

AWS CLI

Pour décrire vos passerelles Internet de sortie uniquement

Cet exemple décrit vos passerelles Internet de sortie uniquement.

Commande :

```
aws ec2 describe-egress-only-internet-gateways
```

Sortie :

```
{
  "EgressOnlyInternetGateways": [
      {
          "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
          "Attachments": [
              {
                  "State": "attached",
                  "VpcId": "vpc-0c62a468"
              }
          ]
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEgressOnlyInternetGatewaysà la section Référence des AWS CLI commandes.

`describe-elastic-gpus`

L'exemple de code suivant montre comment utiliser`describe-elastic-gpus`.

AWS CLI

Pour décrire un GPU élastique

Commande :

```
aws ec2 describe-elastic-gpus --elastic-gpu-
ids egpu-12345678901234567890abcdefghijkl
```

- Pour plus de détails sur l'API, reportez-vous DescribeElasticGpusà la section Référence des AWS CLI commandes.

`describe-export-image-tasks`

L'exemple de code suivant montre comment utiliser`describe-export-image-tasks`.

AWS CLI

Pour surveiller une tâche d'exportation d'image

L'`describe-export-image-tasks`exemple suivant vérifie le statut de la tâche d'exportation d'image spécifiée. Le fichier image obtenu dans Amazon S3 est`my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk`.

```
aws ec2 describe-export-image-tasks \
    --export-image-task-ids export-ami-1234567890abcdef0
```

Sortie pour une tâche d'exportation d'image en cours.

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "Progress": "21",
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "active",
            "StatusMessage": "updating"
        }
    ]
}
```

Sortie pour une tâche d'exportation d'image terminée.

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "completed"
        }
    ]
}
```

Pour plus d'informations, consultez la section Exporter une machine virtuelle depuis une AMI dans le guide de l' Import/Export utilisateur de la machine virtuelle.

- Pour plus de détails sur l'API, reportez-vous DescribeExportImageTasksà la section Référence des AWS CLI commandes.

`describe-export-tasks`

L'exemple de code suivant montre comment utiliser`describe-export-tasks`.

AWS CLI

Pour répertorier les détails d'une tâche d'exportation d'instance

Cet exemple décrit la tâche d'exportation portant l'ID export-i-fh 8sjjsq.

Commande :

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

Sortie :

```
{
    "ExportTasks": [
        {
            "State": "active",
            "InstanceExportDetails": {
                "InstanceId": "i-1234567890abcdef0",
```

```
                "TargetEnvironment": "vmware"
            },
            "ExportToS3Task": {
                "S3Bucket": "myexportbucket",
                "S3Key": "RHEL5export-i-fh8sjjsq.ova",
                "DiskImageFormat": "vmdk",
                "ContainerFormat": "ova"
            },
            "Description": "RHEL5 instance",
            "ExportTaskId": "export-i-fh8sjjsq"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeExportTasksà la section Référence des AWS CLI commandes.

`describe-fast-launch-images`

L'exemple de code suivant montre comment utiliser`describe-fast-launch-images`.

AWS CLI

Pour décrire les informations relatives aux systèmes Windows AMIs configurés pour un lancement plus rapide

L'`describe-fast-launch-images`exemple suivant décrit les détails de chacun des éléments de votre compte configurés pour un lancement plus rapide, notamment le type de ressource, la configuration du snapshot, les détails du modèle de lancement, le nombre maximum de lancements parallèles, l'ID du propriétaire de l'AMI, l'état de la configuration de lancement rapide, la raison pour laquelle l'état a été modifié et l'heure à laquelle le changement d'état s'est produit. AMIs

```
aws ec2 describe-fast-launch-images
```

Sortie :

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
```

```
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
    ]
}
```

Pour plus d'informations sur la configuration d'une AMI Windows pour un lancement plus rapide, consultez Configurer votre AMI pour un lancement plus rapide dans le Guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeFastLaunchImagesà la section Référence des AWS CLI commandes.

`describe-fast-snapshot-restores`

L'exemple de code suivant montre comment utiliser`describe-fast-snapshot-restores`.

AWS CLI

Pour décrire les restaurations rapides de snapshots

L'`describe-fast-snapshot-restores`exemple suivant affiche les détails de toutes les restaurations rapides de snapshots dont l'état est égal à`disabled`.

```
aws ec2 describe-fast-snapshot-restores \
    --filters Name=state,Values=disabled
```

Sortie :

```
{
```

```
    "FastSnapshotRestores": [
        {
            "SnapshotId": "snap-1234567890abcdef0",
            "AvailabilityZone": "us-west-2c",
            "State": "disabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state
 transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z",
            "DisablingTime": "2020-01-26T00:40:56.069Z",
            "DisabledTime": "2020-01-26T00:41:27.390Z"
        }
    ]
}
```

L'`describe-fast-snapshot-restores`exemple suivant décrit toutes les restaurations rapides de snapshots.

```
aws ec2 describe-fast-snapshot-restores
```

- Pour plus de détails sur l'API, reportez-vous DescribeFastSnapshotRestoresà la section Référence des AWS CLI commandes.

`describe-fleet-history`

L'exemple de code suivant montre comment utiliser`describe-fleet-history`.

AWS CLI

Pour décrire l'historique EC2 de la flotte

L'`describe-fleet-history`exemple suivant renvoie l'historique de la EC2 flotte spécifiée à partir de l'heure spécifiée. Le résultat correspond à une EC2 flotte avec deux instances en cours d'exécution.

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --start-time 2020-09-01T00:00:00Z
```

Sortie :


```
{
    "HistoryRecords": [
        {
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:05.000Z"
        },
        {
            "EventInformation": {
                "EventSubType": "active"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:15.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
                "EventSubType": "progress"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-083a1c446e66085d2"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-090db02406cc3c2d6"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        }
```

```
    ],
    "LastEvaluatedTime": "2020-09-01T19:10:19.000Z",
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
    "StartTime": "2020-08-31T23:53:20.000Z"
}
```

Pour plus d'informations, consultez la section Gestion d'un EC2 parc dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetHistoryà la section Référence des AWS CLI commandes.

`describe-fleet-instances`

L'exemple de code suivant montre comment utiliser`describe-fleet-instances`.

AWS CLI

Pour décrire les instances en cours d'exécution d'une EC2 flotte

L'`describe-fleet-instances`exemple suivant décrit les instances en cours d'exécution pour le EC2 parc spécifié.

```
aws ec2 describe-fleet-instances \
    --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

Sortie :

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-090db02406cc3c2d6",
            "InstanceType": "t2.small",
            "SpotInstanceRequestId": "sir-a43gtpfk",
            "InstanceHealth": "healthy"
        },
        {
            "InstanceId": "i-083a1c446e66085d2",
            "InstanceType": "t2.small",
            "SpotInstanceRequestId": "sir-iwcit2nj",
            "InstanceHealth": "healthy"
        }
```

```
    ],
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Pour plus d'informations, consultez la section Gestion d'un EC2 parc dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetInstancesà la section Référence des AWS CLI commandes.

`describe-fleets`

L'exemple de code suivant montre comment utiliser`describe-fleets`.

AWS CLI

Pour décrire une EC2 flotte

L'`describe-fleets`exemple suivant décrit la EC2 flotte spécifiée.

```
aws ec2 describe-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

Sortie :

```
{
    "Fleets": [
        {
            "ActivityStatus": "pending_fulfillment",
            "CreateTime": "2020-09-01T18:26:05.000Z",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "FulfilledCapacity": 0.0,
            "FulfilledOnDemandCapacity": 0.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateId": "lt-0e632f2855a979cd5",
                        "Version": "1"
                    }
                }
```

```
            ],
            "TargetCapacitySpecification": {
                "TotalTargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "TerminateInstancesWithExpiration": false,
            "Type": "maintain",
            "ReplaceUnhealthyInstances": false,
            "SpotOptions": {
                "AllocationStrategy": "lowestPrice",
                "InstanceInterruptionBehavior": "terminate",
                "InstancePoolsToUseCount": 1
            },
            "OnDemandOptions": {
                "AllocationStrategy": "lowestPrice"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion d'un EC2 parc dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetsà la section Référence des AWS CLI commandes.

`describe-flow-logs`

L'exemple de code suivant montre comment utiliser`describe-flow-logs`.

AWS CLI

Exemple 1 : pour décrire tous vos journaux de flux

L'`describe-flow-logs`exemple suivant affiche les détails de tous vos journaux de flux.

```
aws ec2 describe-flow-logs
```

Sortie :

```
{
```

```
    "FlowLogs": [
        {
            "CreationTime": "2018-02-21T13:22:12.644Z",
            "DeliverLogsPermissionArn": "arn:aws:iam::123456789012:role/flow-logs-
role",
            "DeliverLogsStatus": "SUCCESS",
            "FlowLogId": "fl-aabbccdd112233445",
            "MaxAggregationInterval": 600,
            "FlowLogStatus": "ACTIVE",
            "LogGroupName": "FlowLogGroup",
            "ResourceId": "subnet-12345678901234567",
            "TrafficType": "ALL",
            "LogDestinationType": "cloud-watch-logs",
            "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr}
 ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end}
 ${action} ${log-status}"
        },
        {
            "CreationTime": "2020-02-04T15:22:29.986Z",
            "DeliverLogsStatus": "SUCCESS",
            "FlowLogId": "fl-01234567890123456",
            "MaxAggregationInterval": 60,
            "FlowLogStatus": "ACTIVE",
            "ResourceId": "vpc-00112233445566778",
            "TrafficType": "ACCEPT",
            "LogDestinationType": "s3",
            "LogDestination": "arn:aws:s3:::my-flow-log-bucket/custom",
            "LogFormat": "${version} ${vpc-id} ${subnet-id} ${instance-id}
 ${interface-id} ${account-id} ${type} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
 ${pkt-srcaddr} ${pkt-dstaddr} ${protocol} ${bytes} ${packets} ${start} ${end}
 ${action} ${tcp-flags} ${log-status}"
        }
    ]
}
```

Exemple 2 : pour décrire un sous-ensemble de vos journaux de flux

L'`describe-flow-logs`exemple suivant utilise un filtre pour afficher les détails uniquement pour les journaux de flux qui se trouvent dans le groupe de CloudWatch journaux spécifié dans Amazon Logs.

```
aws ec2 describe-flow-logs \
    --filter "Name=log-group-name,Values=MyFlowLogs"
```

- Pour plus de détails sur l'API, reportez-vous DescribeFlowLogsà la section Référence des AWS CLI commandes.

`describe-fpga-image-attribute`

L'exemple de code suivant montre comment utiliser`describe-fpga-image-attribute`.

AWS CLI

Pour décrire les attributs d'une image Amazon FPGA

Cet exemple décrit les autorisations de chargement pour l'AFI spécifié.

Commande :

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

Sortie :

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
          }
      ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeFpgaImageAttributeà la section Référence des AWS CLI commandes.

`describe-fpga-images`

L'exemple de code suivant montre comment utiliser`describe-fpga-images`.

AWS CLI

Pour décrire les images Amazon FPGA

Cet exemple décrit AFIs ceux qui sont détenus par compte123456789012.

Commande :

```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

Sortie :

```
{
  "FpgaImages": [
      {
          "UpdateTime": "2017-12-22T12:09:14.000Z",
          "Name": "my-afi",
          "PciId": {
              "SubsystemVendorId": "0xfedd",
              "VendorId": "0x1d0f",
              "DeviceId": "0xf000",
              "SubsystemId": "0x1d51"
          },
          "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc",
          "Public": false,
          "State": {
              "Code": "available"
          },
          "ShellVersion": "0x071417d3",
          "OwnerId": "123456789012",
          "FpgaImageId": "afi-0d123e123bfc85abc",
          "CreateTime": "2017-12-22T11:43:33.000Z",
          "Description": "my-afi"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeFpgaImagesà la section Référence des AWS CLI commandes.

`describe-host-reservation-offerings`

L'exemple de code suivant montre comment utiliserdescribe-host-reservation-offerings.

AWS CLI

Pour décrire les offres de réservation d'hôtes dédiés

Cet exemple décrit les réservations d'hôtes dédiés pour la famille d'instances M4 disponibles à l'achat.

Commande :

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

Sortie :

```
{
  "OfferingSet": [
      {
          "HourlyPrice": "1.499",
          "OfferingId": "hro-03f707bf363b6b324",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      },
      {
          "HourlyPrice": "1.045",
          "OfferingId": "hro-0ef9181cabdef7a02",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "UpfrontPrice": "0.000",
          "Duration": 94608000
      },
      {
          "HourlyPrice": "0.714",
          "OfferingId": "hro-04567a15500b92a51",
          "InstanceFamily": "m4",
          "PaymentOption": "PartialUpfront",
          "UpfrontPrice": "6254.000",
          "Duration": 31536000
      },
      {
          "HourlyPrice": "0.484",
          "OfferingId": "hro-0d5d7a9d23ed7fbfe",
          "InstanceFamily": "m4",
```

```
            "PaymentOption": "PartialUpfront",
            "UpfrontPrice": "12720.000",
            "Duration": 94608000
        },
        {
            "HourlyPrice": "0.000",
            "OfferingId": "hro-05da4108ca998c2e5",
            "InstanceFamily": "m4",
            "PaymentOption": "AllUpfront",
            "UpfrontPrice": "23913.000",
            "Duration": 94608000
        },
        {
            "HourlyPrice": "0.000",
            "OfferingId": "hro-0a9f9be3b95a3dc8f",
            "InstanceFamily": "m4",
            "PaymentOption": "AllUpfront",
            "UpfrontPrice": "12257.000",
            "Duration": 31536000
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeHostReservationOfferingsà la section Référence des AWS CLI commandes.

`describe-host-reservations`

L'exemple de code suivant montre comment utiliser`describe-host-reservations`.

AWS CLI

Pour décrire les réservations d'hôtes dédiés dans votre compte

Cet exemple décrit les réservations d'hôtes dédiés dans votre compte.

Commande :

```
aws ec2 describe-host-reservations
```

Sortie :

```
{
```

```
  "HostReservationSet": [
      {
          "Count": 1,
          "End": "2019-01-10T12:14:09Z",
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "OfferingId": "hro-03f707bf363b6b324",
          "PaymentOption": "NoUpfront",
          "State": "active",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "Start": "2018-01-10T12:14:09Z",
          "HostReservationId": "hr-0d418a3a4ffc669ae",
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeHostReservationsà la section Référence des AWS CLI commandes.

`describe-hosts`

L'exemple de code suivant montre comment utiliser`describe-hosts`.

AWS CLI

Pour afficher des informations sur les hôtes dédiés

L'`describe-hosts`exemple suivant affiche les détails des hôtes `available` dédiés de votre AWS compte.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

Sortie :

```
{
    "Hosts": [
        {
```

```
            "HostId": "h-07879acf49EXAMPLE",
            "Tags": [
                {
                    "Value": "production",
                    "Key": "purpose"
                }
            ],
            "HostProperties": {
                "Cores": 48,
                "TotalVCpus": 96,
                "InstanceType": "m5.large",
                "Sockets": 2
            },
            "Instances": [],
            "State": "available",
            "AvailabilityZone": "eu-west-1a",
            "AvailableCapacity": {
                "AvailableInstanceCapacity": [
                    {
                        "AvailableCapacity": 48,
                        "InstanceType": "m5.large",
                        "TotalCapacity": 48
                    }
                ],
                "AvailableVCpus": 96
            },
            "HostRecovery": "on",
            "AllocationTime": "2019-08-19T08:57:44.000Z",
            "AutoPlacement": "off"
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation des hôtes dédiés dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeHostsà la section Référence des AWS CLI commandes.

`describe-iam-instance-profile-associations`

L'exemple de code suivant montre comment utiliser`describe-iam-instance-profile-associations`.

AWS CLI

Description des associations de profils d'instances IAM

Cet exemple décrit toutes vos associations de profils d'instance IAM.

Commande :

```
aws ec2 describe-iam-instance-profile-associations
```

Sortie :

```
{
  "IamInstanceProfileAssociations": [
      {
          "InstanceId": "i-09eb09efa73ec1dee",
          "State": "associated",
          "AssociationId": "iip-assoc-0db249b1f25fa24b8",
          "IamInstanceProfile": {
              "Id": "AIPAJVQN4F5WVLGCJDRGM",
              "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
          }
      },
      {
          "InstanceId": "i-0402909a2f4dffd14",
          "State": "associating",
          "AssociationId": "iip-assoc-0d1ec06278d29f44a",
          "IamInstanceProfile": {
              "Id": "AGJAJVQN4F5WVLGCJABCM",
              "Arn": "arn:aws:iam::123456789012:instance-profile/user1-role"
          }
      }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeIamInstanceProfileAssociationsà la section Référence des AWS CLI commandes.

`describe-id-format`

L'exemple de code suivant montre comment utiliser`describe-id-format`.

AWS CLI

Exemple 1 : pour décrire le format d'identification d'une ressource

L'`describe-id-format`exemple suivant décrit le format d'identification des groupes de sécurité.

```
aws ec2 describe-id-format \
    --resource security-group
```

Dans l'exemple de sortie suivant, la `Deadline` valeur indique que la date limite pour que ce type de ressource passe définitivement du format d'identifiant court au format d'identifiant long a expiré à 00h00 UTC le 15 août 2018.

```
{
    "Statuses": [
        {
            "Deadline": "2018-08-15T00:00:00.000Z",
            "Resource": "security-group",
            "UseLongIds": true
        }
    ]
}
```

Exemple 2 : pour décrire le format d'identification de toutes les ressources

L'`describe-id-format`exemple suivant décrit le format d'ID pour tous les types de ressources. Tous les types de ressources prenant en charge le format d'identifiant court ont été changés pour utiliser le format d'identifiant long.

```
aws ec2 describe-id-format
```

- Pour plus de détails sur l'API, reportez-vous DescribeIdFormatà la section Référence des AWS CLI commandes.

`describe-identity-id-format`

L'exemple de code suivant montre comment utiliser`describe-identity-id-format`.

AWS CLI

Pour décrire le format d'ID d'un rôle IAM

L'`describe-identity-id-format`exemple suivant décrit le format d'ID reçu par les instances créées par le rôle IAM `EC2Role` dans votre AWS compte.

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/my-iam-role \
    --resource instance
```

Le résultat suivant indique que les instances créées par ce rôle sont IDs reçues au format ID long.

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "instance",
            "UseLongIds": true
        }
    ]
}
```

Pour décrire le format d'identification d'un utilisateur IAM

L'`describe-identity-id-format`exemple suivant décrit le format d'identification reçu par les instantanés créés par l'utilisateur IAM `AdminUser` dans votre AWS compte.

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource snapshot
```

La sortie indique que les instantanés créés par cet utilisateur sont reçus IDs au format ID long.

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "snapshot",
            "UseLongIds": true
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeIdentityIdFormatà la section Référence des AWS CLI commandes.

`describe-image-attribute`

L'exemple de code suivant montre comment utiliser`describe-image-attribute`.

AWS CLI

Pour décrire les autorisations de lancement d'une AMI

Cet exemple décrit les autorisations de lancement pour l'AMI spécifiée.

Commande :

```
aws ec2 describe-image-attribute --image-id ami-5731123e --
attribute launchPermission
```

Sortie :

```
{
    "LaunchPermissions": [
        {
            "UserId": "123456789012"
        }
    ],
    "ImageId": "ami-5731123e",
}
```

Pour décrire les codes de produit pour une AMI

Cet exemple décrit les codes de produit pour l'AMI spécifiée. Notez que cette AMI ne contient aucun code produit.

Commande :

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

Sortie :

```
{
    "ProductCodes": [],
```

```
    "ImageId": "ami-5731123e",
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeImageAttributeà la section Référence des AWS CLI commandes.

`describe-images`

L'exemple de code suivant montre comment utiliser`describe-images`.

AWS CLI

Exemple 1 : description d'une AMI

L'exemple `describe-images` suivant décrit l'AMI spécifiée dans la région spécifiée.

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-ids ami-1234567890EXAMPLE
```

Sortie :

```
{
    "Images": [
        {
            "VirtualizationType": "hvm",
            "Description": "Provided by Red Hat, Inc.",
            "PlatformDetails": "Red Hat Enterprise Linux",
            "EnaSupport": true,
            "Hypervisor": "xen",
            "State": "available",
            "SriovNetSupport": "simple",
            "ImageId": "ami-1234567890EXAMPLE",
            "UsageOperation": "RunInstances:0010",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "SnapshotId": "snap-111222333444aaabb",
                        "DeleteOnTermination": true,
                        "VolumeType": "gp2",
                        "VolumeSize": 10,
```

```
                    "Encrypted": false
                }
            }
        ],
        "Architecture": "x86_64",
        "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-
GP2",
        "RootDeviceType": "ebs",
        "OwnerId": "123456789012",
        "RootDeviceName": "/dev/sda1",
        "CreationDate": "2019-05-10T13:17:12.000Z",
        "Public": true,
        "ImageType": "machine",
        "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
    }
]
}
```

Pour plus d'informations, consultez Amazon Machine Images (AMI) dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour décrire AMIs en fonction de filtres

L'`describe-images`exemple suivant décrit les systèmes Windows AMIs fournis par Amazon et soutenus par Amazon EBS.

```
aws ec2 describe-images \
    --owners amazon \
    --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"
```

Pour obtenir un exemple de la manière de procéder pour `describe-images`, reportez-vous à l'exemple 1.

Pour d'autres exemples d'utilisation de filtres, consultez la section Répertorier et filtrer vos ressources dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour décrire AMIs en fonction de balises

L'`describe-images`exemple suivant décrit tous ceux AMIs qui possèdent cette baliseType=Custom. L'exemple utilise le `--query` paramètre pour afficher uniquement l'AMI IDs.

```
aws ec2 describe-images \
```

```
    --filters "Name=tag:Type,Values=Custom" \
    --query 'Images[*].[ImageId]' \
    --output text
```

Sortie :

```
ami-1234567890EXAMPLE
ami-0abcdef1234567890
```

Pour d'autres exemples d'utilisation de filtres de balises, consultez la section Utilisation des balises dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeImagesà la section Référence des AWS CLI commandes.

`describe-import-image-tasks`

L'exemple de code suivant montre comment utiliser`describe-import-image-tasks`.

AWS CLI

Pour surveiller une tâche d'importation d'image

L'`describe-import-image-tasks`exemple suivant vérifie le statut de la tâche d'importation d'image spécifiée.

```
aws ec2 describe-import-image-tasks \
    --import-task-ids import-ami-1234567890abcdef0
```

Sortie pour une tâche d'importation d'image en cours.

```
{
    "ImportImageTasks": [
        {
            "ImportTaskId": "import-ami-1234567890abcdef0",
            "Progress": "28",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 705638400.0,
                    "Format": "ova",
                    "Status": "completed",
                    "UserBucket": {
```

```
                    "S3Bucket": "my-import-bucket",
                    "S3Key": "vms/my-server-vm.ova"
                }
            }
        ],
        "Status": "active",
        "StatusMessage": "converting"
    }
]
}
```

Sortie pour une tâche d'importation d'image terminée. L'ID de l'AMI qui en résulte est fourni par`ImageId`.

```
{
    "ImportImageTasks": [
        {
            "ImportTaskId": "import-ami-1234567890abcdef0",
            "ImageId": "ami-1234567890abcdef0",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 705638400.0,
                    "Format": "ova",
                    "SnapshotId": "snap-1234567890abcdef0"
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "my-import-bucket",
                        "S3Key": "vms/my-server-vm.ova"
                    }
                }
            ],
            "Status": "completed"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeImportImageTasksà la section Référence des AWS CLI commandes.

`describe-import-snapshot-tasks`

L'exemple de code suivant montre comment utiliser`describe-import-snapshot-tasks`.

AWS CLI

Pour surveiller une tâche d'importation de snapshots

L'`describe-import-snapshot-tasks`exemple suivant vérifie le statut de la tâche de capture instantanée d'importation spécifiée.

```
aws ec2 describe-import-snapshot-tasks \
    --import-task-ids import-snap-1234567890abcdef0
```

Résultat pour une tâche d'importation de snapshots en cours :

```
{
    "ImportSnapshotTasks": [
        {
            "Description": "My server VMDK",
            "ImportTaskId": "import-snap-1234567890abcdef0",
            "SnapshotTaskDetail": {
                "Description": "My server VMDK",
                "DiskImageSize": "705638400.0",
                "Format": "VMDK",
                "Progress": "42",
                "Status": "active",
                "StatusMessage": "downloading/converting",
                "UserBucket": {
                    "S3Bucket": "my-import-bucket",
                    "S3Key": "vms/my-server-vm.vmdk"
                }
            }
        }
    ]
}
```

Sortie pour une tâche d'importation d'instantanés terminée. L'ID de l'instantané obtenu est fourni par`SnapshotId`.

```
{
    "ImportSnapshotTasks": [
        {
            "Description": "My server VMDK",
            "ImportTaskId": "import-snap-1234567890abcdef0",
            "SnapshotTaskDetail": {
```

```
                "Description": "My server VMDK",
                "DiskImageSize": "705638400.0",
                "Format": "VMDK",
                "SnapshotId": "snap-1234567890abcdef0"
                "Status": "completed",
                "UserBucket": {
                    "S3Bucket": "my-import-bucket",
                    "S3Key": "vms/my-server-vm.vmdk"
                }
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeImportSnapshotTasksà la section Référence des AWS CLI commandes.

`describe-instance-attribute`

L'exemple de code suivant montre comment utiliser`describe-instance-attribute`.

AWS CLI

Pour décrire le type d'instance

Cet exemple décrit le type d'instance de l'instance spécifiée.

Commande :

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute instanceType
```

Sortie :

```
{
    "InstanceId": "i-1234567890abcdef0"
    "InstanceType": {
        "Value": "t1.micro"
    }
}
```

Pour décrire l' disableApiTermination attribut

Cet exemple décrit l'`disableApiTermination`attribut de l'instance spécifiée.

Commande :

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute disableApiTermination
```

Sortie :

```
{
"InstanceId": "i-1234567890abcdef0"
    "DisableApiTermination": {
        "Value": "false"
    }
}
```

Pour décrire le mappage des périphériques en mode bloc pour une instance

Cet exemple décrit l'`blockDeviceMapping`attribut de l'instance spécifiée.

Commande :

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute blockDeviceMapping
```

Sortie :

```
{
    "InstanceId": "i-1234567890abcdef0"
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/sda1",
            "Ebs": {
                "Status": "attached",
                "DeleteOnTermination": true,
                "VolumeId": "vol-049df61146c4d7901",
                "AttachTime": "2013-05-17T22:42:34.000Z"
            }
        },
        {
            "DeviceName": "/dev/sdf",
            "Ebs": {
```

```
                "Status": "attached",
                "DeleteOnTermination": false,
                "VolumeId": "vol-049df61146c4d7901",
                "AttachTime": "2013-09-10T23:07:00.000Z"
            }
        }
    ],
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceAttributeà la section Référence des AWS CLI commandes.

`describe-instance-connect-endpoints`

L'exemple de code suivant montre comment utiliser`describe-instance-connect-endpoints`.

AWS CLI

Pour décrire un point de terminaison EC2 Instance Connect

L'`describe-instance-connect-endpoints`exemple suivant décrit le point de terminaison EC2 Instance Connect spécifié.

```
aws ec2 describe-instance-connect-endpoints \
    --region us-east-1 \
    --instance-connect-endpoint-ids eice-0123456789example
```

Sortie :

```
{
    "InstanceConnectEndpoints": [
        {
            "OwnerId": "111111111111",
            "InstanceConnectEndpointId": "eice-0123456789example",
            "InstanceConnectEndpointArn": "arn:aws:ec2:us-
east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
            "State": "create-complete",
            "StateMessage": "",
            "DnsName": "eice-0123456789example.b67b86ba.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
            "NetworkInterfaceIds": [
                "eni-0123456789example"
```

```
            ],
            "VpcId": "vpc-0123abcd",
            "AvailabilityZone": "us-east-1d",
            "CreatedAt": "2023-02-07T12:05:37+00:00",
            "SubnetId": "subnet-0123abcd",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez Create an EC2 Instance Connect Endpoint dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceConnectEndpointsà la section Référence des AWS CLI commandes.

`describe-instance-credit-specifications`

L'exemple de code suivant montre comment utiliser`describe-instance-credit-specifications`.

AWS CLI

Pour décrire l'option de crédit pour l'utilisation du processeur d'une ou de plusieurs instances

L'`describe-instance-credit-specifications`exemple suivant décrit l'option de crédit du processeur pour l'instance spécifiée.

```
aws ec2 describe-instance-credit-specifications \
    --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
    "InstanceCreditSpecifications": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CpuCredits": "unlimited"
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des instances de performance éclatantes dans le guide de l' EC2 utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceCreditSpecificationsà la section Référence des AWS CLI commandes.

`describe-instance-event-notification-attributes`

L'exemple de code suivant montre comment utiliser`describe-instance-event-notification-attributes`.

AWS CLI

Pour décrire les balises pour les notifications d'événements planifiés

L'`describe-instance-event-notification-attributes`exemple suivant décrit les balises devant apparaître dans les notifications d'événements planifiés.

```
aws ec2 describe-instance-event-notification-attributes
```

Sortie :

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

Pour plus d'informations, consultez la section Événements planifiés pour vos instances dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceEventNotificationAttributesà la section Référence des AWS CLI commandes.

`describe-instance-event-windows`

L'exemple de code suivant montre comment utiliser`describe-instance-event-windows`.

AWS CLI

Exemple 1 : pour décrire toutes les fenêtres d'événements

L'`describe-instance-event-windows`exemple suivant décrit toutes les fenêtres d'événements de la région spécifiée.

```
aws ec2 describe-instance-event-windows \
    --region us-east-1
```

Sortie :

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0abcdef1234567890",
            "Name": "myEventWindowName",
            "CronExpression": "* 21-23 * * 2,3",
            "AssociationTarget": {
                "InstanceIds": [
                    "i-1234567890abcdef0",
                    "i-0598c7d356eba48d7"
                ],
                "Tags": [],
                "DedicatedHostIds": []
            },
            "State": "active",
            "Tags": []
        }

        ...

    ],
    "NextToken": "9d624e0c-388b-4862-a31e-a85c64fc1d4a"
}
```

Exemple 2 : pour décrire une fenêtre d'événements spécifique

L'`describe-instance-event-windows`exemple suivant décrit un événement spécifique en utilisant le `instance-event-window` paramètre pour décrire une fenêtre d'événements spécifique.

```
aws ec2 describe-instance-event-windows \
    --region us-east-1 \
    --instance-event-window-ids iew-0abcdef1234567890
```

Sortie :

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0abcdef1234567890",
            "Name": "myEventWindowName",
            "CronExpression": "* 21-23 * * 2,3",
            "AssociationTarget": {
                "InstanceIds": [
                    "i-1234567890abcdef0",
                    "i-0598c7d356eba48d7"
                ],
                "Tags": [],
                "DedicatedHostIds": []
            },
            "State": "active",
            "Tags": []
        }
}
```

Exemple 3 : pour décrire les fenêtres d'événements correspondant à un ou plusieurs filtres

L'`describe-instance-event-windows`exemple suivant décrit les fenêtres d'événements qui correspondent à un ou plusieurs filtres à l'aide du `filter` paramètre. Le `instance-id` filtre est utilisé pour décrire toutes les fenêtres d'événements associées à l'instance spécifiée. Quand un filtre est utilisé, il recherche une correspondance directe. Cependant, le filtre `instance-id` est différent. S'il n'y a pas de correspondance directe avec l'ID d'instance, cela revient à des associations indirectes avec la fenêtre d'événements, telles que les balises de l'instance ou l'ID d'hôte dédié (si l'instance est un hôte dédié).

```
aws ec2 describe-instance-event-windows \
    --region us-east-1 \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --max-results 100 \
    --next-token <next-token-value>
```

Sortie :

```
{
    "InstanceEventWindows": [
```

```
        {
            "InstanceEventWindowId": "iew-0dbc0adb66f235982",
            "TimeRanges": [
                {
                    "StartWeekDay": "sunday",
                    "StartHour": 2,
                    "EndWeekDay": "sunday",
                    "EndHour": 8
                }
            ],
            "Name": "myEventWindowName",
            "AssociationTarget": {
                "InstanceIds": [],
                "Tags": [],
                "DedicatedHostIds": [
                    "h-0140d9a7ecbd102dd"
                ]
            },
            "State": "active",
            "Tags": []
        }
    ]
}
```

Dans l'exemple de sortie, l'instance se trouve sur un hôte dédié, qui est associé à la fenêtre d'événements.

Pour connaître les contraintes liées à la fenêtre d'événements, consultez les considérations du guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceEventWindowsà la section Référence des AWS CLI commandes.

`describe-instance-image-metadata`

L'exemple de code suivant montre comment utiliser`describe-instance-image-metadata`.

AWS CLI

Exemple 1 : pour décrire les métadonnées de l'AMI pour toutes les instances

L'`describe-instance-image-metadata`exemple suivant décrit les métadonnées AMI de toutes les instances de votre AWS compte dans la région spécifiée.

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1
```

Sortie :

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ],
    "NextToken": "...EXAMPLEwIAABAA2JHaFxLnEXAMPLE..."
}
```

Pour plus d'informations, consultez Amazon Machine Images sur Amazon EC2 dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour décrire les métadonnées de l'AMI pour les instances spécifiées

L'`describe-instance-image-metadata`exemple suivant décrit les métadonnées de l'AMI pour les instances spécifiées.

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1 \
    --instance-ids i-1234567890EXAMPLE i-0987654321EXAMPLE
```

Sortie :

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        },
        {
            "InstanceId": "i-0987654321EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
```

```
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Machine Images sur Amazon EC2 dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour décrire les métadonnées de l'AMI pour les instances basées sur des filtres

L'`describe-instance-image-metadata`exemple suivant décrit les métadonnées de l'AMI `t2.nano` et les `t2.micro` instances de la zone de `us-east-1a` disponibilité.

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1 \
    --filters Name=availability-zone,Values=us-east-1a Name=instance-
type,Values=t2.nano,t2.micro
```

Sortie :

```
{
    "InstanceImageMetadata": [
        {
```

```
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        },
        {
            "InstanceId": "i-0987654321EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
```

```
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ],
    "NextToken": "...EXAMPLEV7ixRYHwIAABAA2JHaFxLnDAzpatfEXAMPLE..."
}
```

Pour plus d'informations, consultez Amazon Machine Images sur Amazon EC2 dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceImageMetadataà la section Référence des AWS CLI commandes.

`describe-instance-status`

L'exemple de code suivant montre comment utiliser`describe-instance-status`.

AWS CLI

Description du statut d'une instance

L'exemple `describe-instance-status` suivant décrit le statut actuel de l'instance spécifiée.

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
    "InstanceStatuses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
```

```
            "AvailabilityZone": "us-east-1d",
            "SystemStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "InstanceStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            }
        }
    ]
}
```

Pour plus d'informations, consultez Surveiller le statut de vos instances dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceStatusà la section Référence des AWS CLI commandes.

`describe-instance-topology`

L'exemple de code suivant montre comment utiliser`describe-instance-topology`.

AWS CLI

Pour décrire la topologie de toutes vos instances

L'`describe-instance-topology`exemple suivant décrit la topologie de toutes vos instances qui correspondent aux types d'instances pris en charge pour cette commande.

```
aws ec2 describe-instance-topology \
    --region us-west-2
```

Sortie :


```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "my-ml-cpg",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
            ],
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
```

```
                "nn-1235301234example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

Pour plus d'informations, y compris d'autres exemples, consultez la topologie des EC2 instances Amazon dans le guide de l' EC2 utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceTopologyà la section Référence des AWS CLI commandes.

`describe-instance-type-offerings`

L'exemple de code suivant montre comment utiliser`describe-instance-type-offerings`.

AWS CLI

Exemple 1 : pour répertorier les types d'instances proposés dans une région

L'`describe-instance-type-offerings`exemple suivant répertorie les types d'instances proposés dans la région configurée comme région par défaut pour la AWS CLI.

```
aws ec2 describe-instance-type-offerings
```

Pour répertorier les types d'instances proposés dans une région différente, spécifiez la région à l'aide du `--region` paramètre.

```
aws ec2 describe-instance-type-offerings \
    --region us-east-2
```

Sortie :

```
{
  "InstanceTypeOfferings": [
      {
          "InstanceType": "m5.2xlarge",
          "LocationType": "region",
          "Location": "us-east-2"
```

```
        },
        {
            "InstanceType": "t3.micro",
            "LocationType": "region",
            "Location": "us-east-2"
        },
        ...
    ]
}
```

Exemple 2 : pour répertorier les types d'instances proposés dans une zone de disponibilité

L'`describe-instance-type-offerings`exemple suivant répertorie les types d'instances proposés dans la zone de disponibilité spécifiée. La zone de disponibilité doit se trouver dans la région spécifiée.

```
aws ec2 describe-instance-type-offerings \
    --location-type availability-zone \
    --filters Name=location,Values=us-east-2a \
    --region us-east-2
```

Exemple 3 : pour vérifier si un type d'instance est pris en charge

La `describe-instance-type-offerings` commande suivante indique si le type d'`c5.xlarge`instance est pris en charge dans la région spécifiée.

```
aws ec2 describe-instance-type-offerings \
    --filters Name=instance-type,Values=c5.xlarge \
    --region us-east-2
```

L'`describe-instance-type-offerings`exemple suivant répertorie tous les types d'instances C5 pris en charge dans la région spécifiée.

```
aws ec2 describe-instance-type-offerings \
    --filters Name=instance-type,Values=c5* \
    --query "InstanceTypeOfferings[].InstanceType" \
    --region us-east-2
```

Sortie :

```
[
```

```
    "c5d.12xlarge",
    "c5d.9xlarge",
    "c5n.xlarge",
    "c5.xlarge",
    "c5d.metal",
    "c5n.metal",
    "c5.large",
    "c5d.2xlarge",
    "c5n.4xlarge",
    "c5.2xlarge",
    "c5n.large",
    "c5n.9xlarge",
    "c5d.large",
    "c5.18xlarge",
    "c5d.18xlarge",
    "c5.12xlarge",
    "c5n.18xlarge",
    "c5.metal",
    "c5d.4xlarge",
    "c5.24xlarge",
    "c5d.xlarge",
    "c5n.2xlarge",
    "c5d.24xlarge",
    "c5.9xlarge",
    "c5.4xlarge"
]
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceTypeOfferingsà la section Référence des AWS CLI commandes.

`describe-instance-types`

L'exemple de code suivant montre comment utiliser`describe-instance-types`.

AWS CLI

Exemple 1 : description d'un type d'instance

L'exemple `describe-instance-types` suivant affiche les détails du type d'instance spécifié.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro
```

Sortie :


```
{
    "InstanceTypes": [
        {
            "InstanceType": "t2.micro",
            "CurrentGeneration": true,
            "FreeTierEligible": true,
            "SupportedUsageClasses": [
                "on-demand",
                "spot"
            ],
            "SupportedRootDeviceTypes": [
                "ebs"
            ],
            "BareMetal": false,
            "Hypervisor": "xen",
            "ProcessorInfo": {
                "SupportedArchitectures": [
                    "i386",
                    "x86_64"
                ],
                "SustainedClockSpeedInGhz": 2.5
            },
            "VCpuInfo": {
                "DefaultVCpus": 1,
                "DefaultCores": 1,
                "DefaultThreadsPerCore": 1,
                "ValidCores": [
                    1
                ],
                "ValidThreadsPerCore": [
                    1
                ]
            },
            "MemoryInfo": {
                "SizeInMiB": 1024
            },
            "InstanceStorageSupported": false,
            "EbsInfo": {
                "EbsOptimizedSupport": "unsupported",
                "EncryptionSupport": "supported"
            },
            "NetworkInfo": {
```

```
                "NetworkPerformance": "Low to Moderate",
                "MaximumNetworkInterfaces": 2,
                "Ipv4AddressesPerInterface": 2,
                "Ipv6AddressesPerInterface": 2,
                "Ipv6Supported": true,
                "EnaSupport": "unsupported"
            },
            "PlacementGroupInfo": {
                "SupportedStrategies": [
                    "partition",
                    "spread"
                ]
            },
            "HibernationSupported": false,
            "BurstablePerformanceSupported": true,
            "DedicatedHostsSupported": false,
            "AutoRecoverySupported": true
        }
    ]
}
```

Pour plus d'informations, consultez les types d'instances dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

Exemple 2 : filtrage des types d'instance disponibles

Vous pouvez définir un filtre pour limiter les résultats aux types d'instances présentant une caractéristique spécifique. L'exemple `describe-instance-types` suivant liste les types d'instance qui prennent en charge l'hibernation.

```
aws ec2 describe-instance-types \
    --filters Name=hibernation-supported,Values=true --query
 'InstanceTypes[*].InstanceType'
```

Sortie :

```
[
    "m5.8xlarge",
    "r3.large",
    "c3.8xlarge",
    "r5.large",
    "m4.4xlarge",
    "c4.large",
```

```
    "m5.xlarge",
    "m4.xlarge",
    "c3.large",
    "c4.8xlarge",
    "c4.4xlarge",
    "c5.xlarge",
    "c5.12xlarge",
    "r5.4xlarge",
    "c5.4xlarge"
]
```

Pour plus d'informations, consultez les types d'instances dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceTypesà la section Référence des AWS CLI commandes.

`describe-instances`

L'exemple de code suivant montre comment utiliser`describe-instances`.

AWS CLI

Exemple 1 : description d'une instance

L'exemple `describe-instances` suivant décrit l'instance spécifiée.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0abcdef1234567890",
                    "InstanceId": "i-1234567890abcdef0",
                    "InstanceType": "t3.nano",
                    "KeyName": "my-key-pair",
```

```
                    "LaunchTime": "2022-11-15T10:48:59+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-2a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
                    "PrivateIpAddress": "10-0-0-157",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                    "PublicIpAddress": "34.253.223.13",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-04a636d18e83cfacb",
                    "VpcId": "vpc-1234567890abcdef0",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2022-11-15T10:49:00+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-02e6ccdca7de29cf2"
                            }
                        }
                    ],
                    "ClientToken": "1234abcd-1234-abcd-1234-d46a8903e9bc",
                    "EbsOptimized": true,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "IamInstanceProfile": {
                        "Arn": "arn:aws:iam::111111111111:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
                        "Id": "11111111111111111111"
                    },
                    "NetworkInterfaces": [
```

```
                        {
                            "Association": {
                                "IpOwnerId": "amazon",
                                "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                                "PublicIp": "34.253.223.13"
                            },
                            "Attachment": {
                                "AttachTime": "2022-11-15T10:48:59+00:00",
                                "AttachmentId": "eni-attach-1234567890abcdefg",
                                "DeleteOnTermination": true,
                                "DeviceIndex": 0,
                                "Status": "attached",
                                "NetworkCardIndex": 0
                            },
                            "Description": "",
                            "Groups": [
                                {
                                    "GroupName": "launch-wizard-146",
                                    "GroupId": "sg-1234567890abcdefg"
                                }
                            ],
                            "Ipv6Addresses": [],
                            "MacAddress": "00:11:22:33:44:55",
                            "NetworkInterfaceId": "eni-1234567890abcdefg",
                            "OwnerId": "104024344472",
                            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                            "PrivateIpAddress": "10-0-0-157",
                            "PrivateIpAddresses": [
                                {
                                    "Association": {
                                        "IpOwnerId": "amazon",
                                        "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                                        "PublicIp": "34.253.223.13"
                                    },
                                    "Primary": true,
                                    "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                                    "PrivateIpAddress": "10-0-0-157"
                                }
                            ],
                            "SourceDestCheck": true,
```

```
                "Status": "in-use",
                "SubnetId": "subnet-1234567890abcdefg",
                "VpcId": "vpc-1234567890abcdefg",
                "InterfaceType": "interface"
            }
        ],
        "RootDeviceName": "/dev/xvda",
        "RootDeviceType": "ebs",
        "SecurityGroups": [
            {
                "GroupName": "launch-wizard-146",
                "GroupId": "sg-1234567890abcdefg"
            }
        ],
        "SourceDestCheck": true,
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-instance"
            }
        ],
        "VirtualizationType": "hvm",
        "CpuOptions": {
            "CoreCount": 1,
            "ThreadsPerCore": 2
        },
        "CapacityReservationSpecification": {
            "CapacityReservationPreference": "open"
        },
        "HibernationOptions": {
            "Configured": false
        },
        "MetadataOptions": {
            "State": "applied",
            "HttpTokens": "optional",
            "HttpPutResponseHopLimit": 1,
            "HttpEndpoint": "enabled",
            "HttpProtocolIpv6": "disabled",
            "InstanceMetadataTags": "enabled"
        },
        "EnclaveOptions": {
            "Enabled": false
        },
        "PlatformDetails": "Linux/UNIX",
```

```
                    "UsageOperation": "RunInstances",
                    "UsageOperationUpdateTime": "2022-11-15T10:48:59+00:00",
                    "PrivateDnsNameOptions": {
                        "HostnameType": "ip-name",
                        "EnableResourceNameDnsARecord": true,
                        "EnableResourceNameDnsAAAARecord": false
                    },
                    "MaintenanceOptions": {
                        "AutoRecovery": "default"
                    }
                }
            ],
            "OwnerId": "111111111111",
            "ReservationId": "r-1234567890abcdefg"
        }
    ]
}
```

Exemple 2 : filtrage des instances correspondant au type spécifié

L'exemple `describe-instances` suivant utilise des filtres pour limiter les résultats aux instances du type spécifié.

```
aws ec2 describe-instances \
    --filters Name=instance-type,Values=m5.large
```

Pour un exemple de sortie, reportez-vous à l'exemple 1.

Pour plus d'informations, consultez Répertorier et filtrer à l'aide de la CLI dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : filtrage des instances correspondant au type et à la zone de disponibilité spécifiés

L'exemple `describe-instances` suivant utilise plusieurs filtres pour limiter les résultats aux instances du type spécifié qui se trouvent également dans la zone de disponibilité spécifiée.

```
aws ec2 describe-instances \
    --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-
zone,Values=us-east-2c
```

Pour un exemple de sortie, reportez-vous à l'exemple 1.

Exemple 4 : filtrage des instances dont le type et la zone de disponibilité sont spécifiés à l'aide d'un fichier JSON

L'exemple `describe-instances` suivant utilise un fichier d'entrée JSON pour effectuer le même filtrage que dans l'exemple précédent. Lorsque les filtres deviennent plus complexes, il peut être plus facile de les spécifier dans un fichier JSON.

```
aws ec2 describe-instances \
    --filters file://filters.json
```

Contenu de `filters.json` :

```
[
    {
        "Name": "instance-type",
        "Values": ["t2.micro", "t3.micro"]
    },
    {
        "Name": "availability-zone",
        "Values": ["us-east-2c"]
    }
]
```

Pour un exemple de sortie, reportez-vous à l'exemple 1.

Exemple 5 : filtrage des instances comportant la balise Owner spécifiée

L'exemple `describe-instances` suivant utilise des filtres de balises pour limiter les résultats aux instances comportant une balise avec la clé de balise spécifiée (Owner), quelle que soit la valeur de la balise.

```
aws ec2 describe-instances \
    --filters "Name=tag-key,Values=Owner"
```

Pour un exemple de sortie, reportez-vous à l'exemple 1.

Exemple 6 : filtrage des instances comportant la valeur de balise my-team spécifiée

L'exemple `describe-instances` suivant utilise des filtres de balises pour limiter les résultats aux instances comportant une balise avec la valeur de balise spécifiée (my-team), quelle que soit la clé de balise.

```
aws ec2 describe-instances \
    --filters "Name=tag-value,Values=my-team"
```

Pour un exemple de sortie, reportez-vous à l'exemple 1.

Exemple 7 : filtrage des instances comportant la balise Owner et la valeur de balise my-team spécifiées

L'exemple `describe-instances` suivant utilise des filtres de balises pour limiter les résultats aux instances comportant la balise spécifiée (Owner=my-team).

```
aws ec2 describe-instances \
    --filters "Name=tag:Owner,Values=my-team"
```

Pour un exemple de sortie, reportez-vous à l'exemple 1.

Exemple 8 : Pour afficher uniquement l'instance et le sous-réseau IDs pour toutes les instances

Les `describe-instances` exemples suivants utilisent le `--query` paramètre pour afficher uniquement l'instance et le sous-réseau de toutes IDs les instances, au format JSON.

Linux et macOS :

```
aws ec2 describe-instances \
    --query 'Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}' \
    --output json
```

Windows :

```
aws ec2 describe-instances ^
    --query "Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}" ^
    --output json
```

Sortie :

```
[
    {
        "Instance": "i-057750d42936e468a",
        "Subnet": "subnet-069beee9b12030077"
```

```
    },
    {
        "Instance": "i-001efd250faaa6ffa",
        "Subnet": "subnet-0b715c6b7db68927a"
    },
    {
        "Instance": "i-027552a73f021f3bd",
        "Subnet": "subnet-0250c25a1f4e15235"
    }
    ...
]
```

Exemple 9 : Pour filtrer les instances du type spécifié et n'afficher que leur instance IDs

L'`describe-instances`exemple suivant utilise des filtres pour étendre les résultats aux instances du type spécifié et le `--query` paramètre pour afficher uniquement l'instance IDs.

```
aws ec2 describe-instances \
    --filters "Name=instance-type,Values=t2.micro" \
    --query "Reservations[*].Instances[*].[InstanceId]" \
    --output text
```

Sortie :

```
i-031c0dc19de2fb70c
i-00d8bff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

Exemple 10 : Pour filtrer les instances du type spécifié et afficher uniquement leur instance IDs, leur zone de disponibilité et la valeur de balise spécifiée

Les exemples `describe-instances` suivants affichent l'ID de l'instance, la zone de disponibilité et la valeur de la balise `Name` pour les instances comportant une balise nommée `tag-key`, sous forme de tableau.

Linux et macOS :

```
aws ec2 describe-instances \
```

```
    --filters Name=tag-key,Values=Name \
    --query 'Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key==`Name`]|
[0].Value}' \
    --output table
```

Windows :

```
aws ec2 describe-instances ^
    --filters Name=tag-key,Values=Name ^
    --query "Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name']|
[0].Value}" ^
    --output table
```

Sortie :

```
-------------------------------------------------------------
|                     DescribeInstances                     |
+--------------+-----------------------+--------------------+
|      AZ      |       Instance        |        Name        |
+--------------+-----------------------+--------------------+
|  us-east-2b  |  i-057750d42936e468a  |  my-prod-server    |
|  us-east-2a  |  i-001efd250faaa6ffa  |  test-server-1     |
|  us-east-2a  |  i-027552a73f021f3bd  |  test-server-2     |
+--------------+-----------------------+--------------------+
```

Exemple 11 : description des instances d'un groupe de placement de partitions

L'exemple `describe-instances` suivant décrit l'instance spécifiée. La réponse inclut les informations sur le placement, notamment le nom du groupe de placement et le nombre de partitions correspondant à l'instance.

```
aws ec2 describe-instances \
    --instance-ids i-0123a456700123456 \
    --query "Reservations[*].Instances[*].Placement"
```

Sortie :

```
[
```

```
    [
        {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 3,
            "Tenancy": "default"
        }

    ]
]
```

Pour plus d'informations, consultez la section Décrire les instances d'un groupe de placement dans le guide de EC2 l'utilisateur Amazon.

Exemple 12 : filtrage des instances correspondant au groupe de placement et au numéro de partition spécifiés

L'exemple `describe-instances` suivant filtre les résultats uniquement sur les instances ayant le groupe de placement et le numéro de partition spécifiés.

```
aws ec2 describe-instances \
    --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-partition-number,Values=7"
```

Seules les informations pertinentes de la sortie sont présentées ci-dessous.

```
"Instances": [
    {
        "InstanceId": "i-0123a456700123456",
        "InstanceType": "r4.large",
        "Placement": {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 7,
            "Tenancy": "default"
        }
    },
    {
        "InstanceId": "i-9876a543210987654",
        "InstanceType": "r4.large",
        "Placement": {
            "AvailabilityZone": "us-east-1c",
```

```
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 7,
            "Tenancy": "default"
        }
    ],
```

Pour plus d'informations, consultez la section Décrire les instances d'un groupe de placement dans le guide de EC2 l'utilisateur Amazon.

Exemple 13 : filtrage des instances configurées pour autoriser l'accès aux balises à partir des métadonnées de l'instance

L'exemple `describe-instances` suivant filtre les résultats uniquement sur les instances configurées pour autoriser l'accès aux balises d'instance à partir des métadonnées de l'instance.

```
aws ec2 describe-instances \
    --filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \
    --query "Reservations[*].Instances[*].InstanceId" \
    --output text
```

Le résultat escompté est le suivant.

```
i-1234567890abcdefg
i-abcdefg1234567890
i-11111111aaaaaaaaa
i-aaaaaaaa111111111
```

Pour plus d'informations, consultez la section Utilisation des balises d'instance dans les métadonnées des instances dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeInstancesà la section Référence des AWS CLI commandes.

`describe-internet-gateways`

L'exemple de code suivant montre comment utiliser`describe-internet-gateways`.

AWS CLI

Pour décrire une passerelle Internet

L'`describe-internet-gateways`exemple suivant décrit la passerelle Internet spécifiée.

```
aws ec2 describe-internet-gateways \
    --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

Sortie :

```
{
    "InternetGateways": [
        {
            "Attachments": [
                {
                    "State": "available",
                    "VpcId": "vpc-0a60eb65b4EXAMPLE"
                }
            ],
            "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
            "OwnerId": "123456789012",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-igw"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Passerelles Internet dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeInternetGatewaysà la section Référence des AWS CLI commandes.

`describe-ipam-pools`

L'exemple de code suivant montre comment utiliser`describe-ipam-pools`.

AWS CLI

Pour consulter les détails d'un pool IPAM

L'`describe-ipam-pools`exemple suivant montre les détails des pools.

(Linux) :

```
aws ec2 describe-ipam-pools \
    --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-scope-02fc38cd4c48e7d38
```

(Fenêtres) :

```
aws ec2 describe-ipam-pools ^
    --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-scope-02fc38cd4c48e7d38
```

Sortie :

```
{
    "IpamPools": [
        {
            "OwnerId": "123456789012",
            "IpamPoolId": "ipam-pool-02ec043a19bbe5d08",
            "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-pool-02ec043a19bbe5d08",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-scope-02fc38cd4c48e7d38",
            "IpamScopeType": "private",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "Locale": "None",
            "PoolDepth": 1,
            "State": "create-complete",
            "AutoImport": true,
            "AddressFamily": "ipv4",
            "AllocationMinNetmaskLength": 16,
            "AllocationMaxNetmaskLength": 26,
            "AllocationDefaultNetmaskLength": 24,
            "AllocationResourceTags": [
                {
                    "Key": "Environment",
                    "Value": "Preprod"
                }
            ],
```

```
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Preprod pool"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeIpamPoolsà la section Référence des AWS CLI commandes.

`describe-ipam-resource-discoveries`

L'exemple de code suivant montre comment utiliser`describe-ipam-resource-discoveries`.

AWS CLI

Exemple 1 : Afficher les détails complets des découvertes de ressources

Dans cet exemple, vous êtes un administrateur IPAM délégué qui souhaite créer et partager une découverte de ressources avec l'administrateur IPAM d'une autre AWS organisation afin que celui-ci puisse gérer et surveiller les adresses IP des ressources de votre organisation.

Cet exemple peut être utile si :

Vous avez essayé de créer une découverte de ressources, mais vous avez reçu un message d'erreur indiquant que vous avez atteint votre limite de 1. Vous vous rendez compte que vous avez peut-être déjà créé une découverte de ressources et que vous souhaitez la consulter dans votre compte. Vous avez des ressources dans une région qui ne sont pas découvertes par l'IPAM. Vous voulez voir la `--operating-regions` définition de la ressource et vous assurer que vous avez ajouté la bonne région en tant que région opérationnelle afin que les ressources puissent être découvertes.

L'`describe-ipam-resource-discoveries`exemple suivant répertorie les détails de la découverte des ressources dans votre AWS compte. Vous pouvez effectuer une découverte de ressources par AWS région.

```
aws ec2 describe-ipam-resource-discoveries \
```

```
    --region us-east-1
```

Sortie :

```
{
    "IpamResourceDiscoveries": [
        {
            "OwnerId": "149977607591",
            "IpamResourceDiscoveryId": "ipam-res-disco-0f8bdee9067137c0d",
            "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0f8bdee9067137c0d",
            "IpamResourceDiscoveryRegion": "us-east-1",
            "OperatingRegions": [
                {
                    "RegionName": "us-east-1"
                }
            ],
            "IsDefault": false,
            "State": "create-complete",
            "Tags": []
    }
]
}
```

Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

Exemple 2 : Afficher uniquement la découverte des ressources IDs

L'`describe-ipam-resource-discoveries`exemple suivant répertorie l'ID de la ressource découverte dans votre AWS compte. Vous pouvez effectuer une découverte de ressources par AWS région.

```
aws ec2 describe-ipam-resource-discoveries \
    --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \
    --output text
```

Sortie :

```
ipam-res-disco-0481e39b242860333
```

Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DescribeIpamResourceDiscoveriesà la section Référence des AWS CLI commandes.

`describe-ipam-resource-discovery-associations`

L'exemple de code suivant montre comment utiliser`describe-ipam-resource-discovery-associations`.

AWS CLI

Pour afficher toutes les associations de découverte de ressources avec votre IPAM

Dans cet exemple, vous êtes un administrateur délégué IPAM qui a associé les découvertes de ressources à votre IPAM pour intégrer d'autres comptes à votre IPAM. Vous avez remarqué que votre IPAM ne découvre pas les ressources dans les régions d'exploitation de la découverte des ressources comme prévu. Vous souhaitez vérifier le statut et l'état de la découverte de ressources pour vous assurer que le compte qui l'a créée est toujours actif et que la découverte de ressources est toujours partagée.

`--region`Il doit s'agir de la région d'origine de votre IPAM.

L'`describe-ipam-resource-discovery-associations`exemple suivant répertorie les associations de découverte de ressources de votre AWS compte.

```
aws ec2 describe-ipam-resource-discovery-associations \
    --region us-east-1
```

Sortie :

```
{
    "IpamResourceDiscoveryAssociations": [
        {
            "OwnerId": "320805250157",
            "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-05e6b45eca5bf5cf7",
            "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-05e6b45eca5bf5cf7",
            "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
```

```
            "IpamId": "ipam-005f921c17ebd5107",
            "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
            "IpamRegion": "us-east-1",
            "IsDefault": true,
            "ResourceDiscoveryStatus": "active",
            "State": "associate-complete",
            "Tags": []
        },
        {
            "OwnerId": "149977607591",
            "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-0dfd21ae189ab5f62",
            "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::149977607591:ipam-
resource-discovery-association/ipam-res-disco-assoc-0dfd21ae189ab5f62",
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "IpamId": "ipam-005f921c17ebd5107",
            "IpamArn": "arn:aws:ec2::149977607591:ipam/ipam-005f921c17ebd5107",
            "IpamRegion": "us-east-1",
            "IsDefault": false,
            "ResourceDiscoveryStatus": "active",
            "State": "create-complete",
            "Tags": []
        }
    ]
}
```

Dans cet exemple, après avoir exécuté cette commande, vous remarquez que vous avez découvert une ressource autre que celle par défaut (`"IsDefault": false ``) that is ``"ResourceDiscoveryStatus": "not-found"et"State": "create-complete"`. Le compte du propriétaire de la recherche de ressources a été fermé. Si, dans un autre cas, vous remarquez que c'est le cas `"ResourceDiscoveryStatus": "not-found"` et `"State": "associate-complete"` que cela indique que l'un des événements suivants s'est produit :

La découverte de ressources a été supprimée par le propriétaire de la découverte de ressources. Le propriétaire de la découverte de ressources a annulé le partage de la découverte de ressources.

Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DescribeIpamResourceDiscoveryAssociationsà la section Référence des AWS CLI commandes.

`describe-ipam-scopes`

L'exemple de code suivant montre comment utiliser`describe-ipam-scopes`.

AWS CLI

Pour afficher les détails d'un scope IPAM

L'`describe-ipam-scopes`exemple suivant montre les détails des portées.

```
aws ec2 describe-ipam-scopes \
    --filters Name=owner-id,Values=123456789012 Name=ipam-
id,Values=ipam-08440e7a3acde3908
```

Sortie :

```
{
    "IpamScopes": [
        {
            "OwnerId": "123456789012",
            "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "IpamScopeType": "private",
            "IsDefault": true,
            "PoolCount": 2,
            "State": "create-complete",
            "Tags": []
        },
        {
            "OwnerId": "123456789012",
            "IpamScopeId": "ipam-scope-0b9eed026396dbc16",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-0b9eed026396dbc16",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "IpamScopeType": "public",
            "IsDefault": true,
            "PoolCount": 0,
            "State": "create-complete",
            "Tags": []
```

```
        },
        {
            "OwnerId": "123456789012",
            "IpamScopeId": "ipam-scope-0f1aff29486355c22",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-0f1aff29486355c22",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "IpamScopeType": "private",
            "IsDefault": false,
            "Description": "Example description",
            "PoolCount": 0,
            "State": "create-complete",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Example name value"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeIpamScopesà la section Référence des AWS CLI commandes.

`describe-ipams`

L'exemple de code suivant montre comment utiliser`describe-ipams`.

AWS CLI

Pour consulter les détails d'un IPAM

L'`describe-ipams`exemple suivant montre les détails d'un IPAM.

```
aws ec2 describe-ipams \
    --filters Name=owner-id,Values=123456789012
```

Sortie :

```
{
```

```
    "Ipams": [
        {
            "OwnerId": "123456789012",
            "IpamId": "ipam-08440e7a3acde3908",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
            "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
            "ScopeCount": 3,
            "OperatingRegions": [
                {
                    "RegionName": "us-east-1"
                },
                {
                    "RegionName": "us-east-2"
                },
                {
                    "RegionName": "us-west-1"
                }
            ],
            "State": "create-complete",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "ExampleIPAM"
                }
            ]
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous DescribeIpamsà la section Référence des AWS CLI commandes.

`describe-ipv6-pools`

L'exemple de code suivant montre comment utiliser`describe-ipv6-pools`.

AWS CLI

Pour décrire vos pools IPv6 d'adresses

L'`describe-ipv6-pools`exemple suivant affiche les détails de tous vos pools d' IPv6 adresses.

```
aws ec2 describe-ipv6-pools
```

Sortie :

```
{
    "Ipv6Pools": [
        {
            "PoolId": "ipv6pool-ec2-012345abc12345abc",
            "PoolCidrBlocks": [
                {
                    "Cidr": "2001:db8:123::/48"
                }
            ],
            "Tags": [
                {
                    "Key": "pool-1",
                    "Value": "public"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeIpv6Pools dans AWS CLI Command Reference.

`describe-key-pairs`

L'exemple de code suivant montre comment utiliser`describe-key-pairs`.

AWS CLI

Affichage d'une paire de clés

L'exemple `describe-key-pairs` suivant affiche des informations sur la paire de clés spécifiée.

```
aws ec2 describe-key-pairs \
    --key-names my-key-pair
```

Sortie :

```
{
    "KeyPairs": [
```

```
        {
            "KeyPairId": "key-0b94643da6EXAMPLE",
            "KeyFingerprint":
 "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
            "KeyName": "my-key-pair",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2022-05-27T21:51:16.000Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Décrire les clés publiques dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeKeyPairsà la section Référence des AWS CLI commandes.

`describe-launch-template-versions`

L'exemple de code suivant montre comment utiliser`describe-launch-template-versions`.

AWS CLI

Pour décrire les versions des modèles de lancement

Cet exemple décrit les versions du modèle de lancement spécifié.

Commande :

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

Sortie :

```
{
  "LaunchTemplateVersions": [
      {
          "LaunchTemplateId": "lt-068f72b72934aff71",
          "LaunchTemplateName": "Webservers",
          "VersionNumber": 3,
          "CreatedBy": "arn:aws:iam::123456789102:root",
          "LaunchTemplateData": {
              "KeyName": "kp-us-east",
```

```
            "ImageId": "ami-6057e21a",
            "InstanceType": "t2.small",
            "NetworkInterfaces": [
                {
                    "SubnetId": "subnet-7b16de0c",
                    "DeviceIndex": 0,
                    "Groups": [
                        "sg-7c227019"
                    ]
                }
            ]
        },
        "DefaultVersion": false,
        "CreateTime": "2017-11-20T13:19:54.000Z"
    },
    {
        "LaunchTemplateId": "lt-068f72b72934aff71",
        "LaunchTemplateName": "Webservers",
        "VersionNumber": 2,
        "CreatedBy": "arn:aws:iam::123456789102:root",
        "LaunchTemplateData": {
            "KeyName": "kp-us-east",
            "ImageId": "ami-6057e21a",
            "InstanceType": "t2.medium",
            "NetworkInterfaces": [
                {
                    "SubnetId": "subnet-1a2b3c4d",
                    "DeviceIndex": 0,
                    "Groups": [
                        "sg-7c227019"
                    ]
                }
            ]
        },
        "DefaultVersion": false,
        "CreateTime": "2017-11-20T13:12:32.000Z"
    },
    {
        "LaunchTemplateId": "lt-068f72b72934aff71",
        "LaunchTemplateName": "Webservers",
        "VersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789102:root",
        "LaunchTemplateData": {
            "UserData": "",
```

```
            "KeyName": "kp-us-east",
            "ImageId": "ami-aabbcc11",
            "InstanceType": "t2.medium",
            "NetworkInterfaces": [
                {
                    "SubnetId": "subnet-7b16de0c",
                    "DeviceIndex": 0,
                    "DeleteOnTermination": false,
                    "Groups": [
                        "sg-7c227019"
                    ],
                    "AssociatePublicIpAddress": true
                }
            ]
        },
        "DefaultVersion": true,
        "CreateTime": "2017-11-20T12:52:33.000Z"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLaunchTemplateVersionsà la section Référence des AWS CLI commandes.

`describe-launch-templates`

L'exemple de code suivant montre comment utiliser`describe-launch-templates`.

AWS CLI

Pour décrire les modèles de lancement

Cet exemple décrit vos modèles de lancement.

Commande :

```
aws ec2 describe-launch-templates
```

Sortie :

```
{
  "LaunchTemplates": [
```

```
      {
          "LatestVersionNumber": 2,
          "LaunchTemplateId": "lt-0e06d290751193123",
          "LaunchTemplateName": "TemplateForWebServer",
          "DefaultVersionNumber": 2,
          "CreatedBy": "arn:aws:iam::123456789012:root",
          "CreateTime": "2017-11-27T09:30:23.000Z"
      },
      {
          "LatestVersionNumber": 6,
          "LaunchTemplateId": "lt-0c45b5e061ec98456",
          "LaunchTemplateName": "DBServersTemplate",
          "DefaultVersionNumber": 1,
          "CreatedBy": "arn:aws:iam::123456789012:root",
          "CreateTime": "2017-11-20T09:25:22.000Z"
      },
      {
          "LatestVersionNumber": 1,
          "LaunchTemplateId": "lt-0d47d774e8e52dabc",
          "LaunchTemplateName": "MyLaunchTemplate2",
          "DefaultVersionNumber": 1,
          "CreatedBy": "arn:aws:iam::123456789012:root",
          "CreateTime": "2017-11-02T12:06:21.000Z"
      },
      {
          "LatestVersionNumber": 3,
          "LaunchTemplateId": "lt-01e5f948eb4f589d6",
          "LaunchTemplateName": "testingtemplate2",
          "DefaultVersionNumber": 1,
          "CreatedBy": "arn:aws:sts::123456789012:assumed-role/AdminRole/
i-03ee35176e2e5aabc",
          "CreateTime": "2017-12-01T08:19:48.000Z"
      },
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLaunchTemplatesà la section Référence des AWS CLI commandes.

`describe-local-gateway-route-table-virtual-interface-group-associations`

L'exemple de code suivant montre comment utiliser`describe-local-gateway-route-table-virtual-interface-group-associations`.

AWS CLI

Pour décrire les associations entre les groupes d'interfaces virtuelles et les tables de routage des passerelles locales

L'`describe-local-gateway-route-table-virtual-interface-group-associations`exemple suivant décrit les associations entre les groupes d'interfaces virtuelles et les tables de routage des passerelles locales dans votre AWS compte.

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

Sortie :

```
{
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociations": [
        {
            "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-
grp-assoc-07145b276bEXAMPLE",
            "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
            "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:123456789012:local-
gateway-route-table/lgw-rtb-059615ef7dEXAMPLE",
            "OwnerId": "123456789012",
            "State": "associated",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des passerelles locales dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociationsà la section Référence des AWS CLI commandes.

`describe-local-gateway-route-table-vpc-associations`

L'exemple de code suivant montre comment utiliser`describe-local-gateway-route-table-vpc-associations`.

AWS CLI

Pour décrire les associations entre les tables de routage des passerelles locales VPCs et les tables de routage

L'`describe-local-gateway-route-table-vpc-associations`exemple suivant affiche des informations sur l'association spécifiée entre VPCs et les tables de routage de passerelle locales.

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
    --local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

Sortie :

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af1EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-0efe9bde08EXAMPLE",
        "State": "associated"
    }
}
```

Pour en savoir plus, consultez Tables de routage de passerelle locale dans le Guide de l'utilisateur Outposts.

- Pour plus de détails sur l'API, reportez-vous DescribeLocalGatewayRouteTableVpcAssociationsà la section Référence des AWS CLI commandes.

`describe-local-gateway-route-tables`

L'exemple de code suivant montre comment utiliser`describe-local-gateway-route-tables`.

AWS CLI

Pour décrire vos tables de routage de passerelle locale

L'`describe-local-gateway-route-tables`exemple suivant affiche des détails sur les tables de routage des passerelles locales.

```
aws ec2 describe-local-gateway-route-tables
```

Sortie :

```
{
    "LocalGatewayRouteTables": [
        {
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
            "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
            "State": "available"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLocalGatewayRouteTablesà la section Référence des AWS CLI commandes.

`describe-local-gateway-virtual-interface-groups`

L'exemple de code suivant montre comment utiliser`describe-local-gateway-virtual-interface-groups`.

AWS CLI

Pour décrire les groupes d'interfaces virtuelles de passerelle locale

L'`describe-local-gateway-virtual-interface-groups`exemple suivant décrit les groupes d'interfaces virtuelles de passerelle locale de votre AWS compte.

```
aws ec2 describe-local-gateway-virtual-interface-groups
```

Sortie :

```
{
    "LocalGatewayVirtualInterfaceGroups": [
        {
            "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
```

```
            "LocalGatewayVirtualInterfaceIds": [
                "lgw-vif-01a23bc4d5EXAMPLE",
                "lgw-vif-543ab21012EXAMPLE"
            ],
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "OwnerId": "123456789012",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des passerelles locales dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DescribeLocalGatewayVirtualInterfaceGroupsà la section Référence des AWS CLI commandes.

`describe-local-gateway-virtual-interfaces`

L'exemple de code suivant montre comment utiliser`describe-local-gateway-virtual-interfaces`.

AWS CLI

Pour décrire les interfaces virtuelles des passerelles locales

L'`describe-local-gateway-virtual-interfaces`exemple suivant décrit les interfaces virtuelles de passerelle locale de votre AWS compte.

```
aws ec2 describe-local-gateway-virtual-interfaces
```

Sortie :

```
{
    "LocalGatewayVirtualInterfaces": [
        {
            "LocalGatewayVirtualInterfaceId": "lgw-vif-01a23bc4d5EXAMPLE",
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "Vlan": 2410,
            "LocalAddress": "0.0.0.0/0",
            "PeerAddress": "0.0.0.0/0",
```

```
            "LocalBgpAsn": 65010,
            "PeerBgpAsn": 65000,
            "OwnerId": "123456789012",
            "Tags": []
        },
        {
            "LocalGatewayVirtualInterfaceId": "lgw-vif-543ab21012EXAMPLE",
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "Vlan": 2410,
            "LocalAddress": "0.0.0.0/0",
            "PeerAddress": "0.0.0.0/0",
            "LocalBgpAsn": 65010,
            "PeerBgpAsn": 65000,
            "OwnerId": "123456789012",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des passerelles locales dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous DescribeLocalGatewayVirtualInterfacesà la section Référence des AWS CLI commandes.

`describe-local-gateways`

L'exemple de code suivant montre comment utiliser`describe-local-gateways`.

AWS CLI

Pour décrire vos passerelles locales

L'`describe-local-gateways`exemple suivant affiche les détails des passerelles locales mises à votre disposition.

```
aws ec2 describe-local-gateways
```

Sortie :

```
{
```

```
    "LocalGateways": [
        {
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0dc11b66ed59f995a",
            "OwnerId": "123456789012",
            "State": "available"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLocalGatewaysà la section Référence des AWS CLI commandes.

`describe-locked-snapshots`

L'exemple de code suivant montre comment utiliser`describe-locked-snapshots`.

AWS CLI

Pour décrire l'état de verrouillage d'un instantané

L'`describe-locked-snapshots`exemple suivant décrit l'état de verrouillage du cliché spécifié.

```
aws ec2 describe-locked-snapshots \
    --snapshot-ids snap-0b5e733b4a8df6e0d
```

Sortie :

```
{
    "Snapshots": [
        {
            "OwnerId": "123456789012",
            "SnapshotId": "snap-0b5e733b4a8df6e0d",
            "LockState": "governance",
            "LockDuration": 365,
            "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",
            "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00",
            "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez Snapshot Lock dans le guide de l'utilisateur d'Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous DescribeLockedSnapshotsà la section Référence des AWS CLI commandes.

`describe-managed-prefix-lists`

L'exemple de code suivant montre comment utiliser`describe-managed-prefix-lists`.

AWS CLI

Pour décrire les listes de préfixes gérées

L'`describe-managed-prefix-lists`exemple suivant décrit les listes de préfixes détenues par AWS compte`123456789012`.

```
aws ec2 describe-managed-prefix-lists \
    --filters Name=owner-id,Values=123456789012
```

Sortie :

```
{
    "PrefixLists": [
        {
            "PrefixListId": "pl-11223344556677aab",
            "AddressFamily": "IPv6",
            "State": "create-complete",
            "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-11223344556677aab",
            "PrefixListName": "vpc-ipv6-cidrs",
            "MaxEntries": 25,
            "Version": 1,
            "Tags": [],
            "OwnerId": "123456789012"
        },
        {
            "PrefixListId": "pl-0123456abcabcabc1",
            "AddressFamily": "IPv4",
            "State": "active",
            "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
```

```
            "PrefixListName": "vpc-cidrs",
            "MaxEntries": 10,
            "Version": 1,
            "Tags": [],
            "OwnerId": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeManagedPrefixListsà la section Référence des AWS CLI commandes.

`describe-moving-addresses`

L'exemple de code suivant montre comment utiliser`describe-moving-addresses`.

AWS CLI

Pour décrire vos adresses de déménagement

Cet exemple décrit toutes vos adresses IP Elastic mobiles.

Commande :

```
aws ec2 describe-moving-addresses
```

Sortie :

```
{
    "MovingAddressStatuses": [
        {
            "PublicIp": "198.51.100.0",
            "MoveStatus": "MovingToVpc"
        }
    ]
}
```

Cet exemple décrit toutes les adresses qui migrent vers la plateforme EC2 -VPC.

Commande :

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- Pour plus de détails sur l'API, reportez-vous DescribeMovingAddressesà la section Référence des AWS CLI commandes.

`describe-nat-gateways`

L'exemple de code suivant montre comment utiliser`describe-nat-gateways`.

AWS CLI

Exemple 1 : pour décrire une passerelle NAT publique

L'`describe-nat-gateways`exemple suivant décrit la passerelle NAT publique spécifiée.

```
aws ec2 describe-nat-gateways \
    --nat-gateway-id nat-01234567890abcdef
```

Sortie :

```
{
    "NatGateways": [
        {
            "CreateTime": "2023-08-25T01:56:51.000Z",
            "NatGatewayAddresses": [
                {
                    "AllocationId": "eipalloc-0790180cd2EXAMPLE",
                    "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
                    "PrivateIp": "10.0.0.211",
                    "PublicIp": "54.85.121.213",
                    "AssociationId": "eipassoc-04d295cc9b8815b24",
                    "IsPrimary": true,
                    "Status": "succeeded"
                },
                {
                    "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
                    "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
                    "PrivateIp": "10.0.0.74",
                    "PublicIp": "3.211.231.218",
                    "AssociationId": "eipassoc-0f96bdca17EXAMPLE",
```

```
                    "IsPrimary": false,
                    "Status": "succeeded"
                }
            ],
            "NatGatewayId": "nat-01234567890abcdef",
            "State": "available",
            "SubnetId": "subnet-655eab5f08EXAMPLE",
            "VpcId": "vpc-098eb5ef58EXAMPLE",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "public-nat"
                }
            ],
            "ConnectivityType": "public"
        }
    ]
}
```

Exemple 2 : pour décrire une passerelle NAT privée

L'`describe-nat-gateways`exemple suivant décrit la passerelle NAT privée spécifiée.

```
aws ec2 describe-nat-gateways \
    --nat-gateway-id nat-1234567890abcdef0
```

Sortie :

```
{
    "NatGateways": [
        {
            "CreateTime": "2023-08-25T00:50:05.000Z",
            "NatGatewayAddresses": [
                {
                    "NetworkInterfaceId": "eni-0065a61b324d1897a",
                    "PrivateIp": "10.0.20.240",
                    "IsPrimary": true,
                    "Status": "succeeded"
                },
                {
                    "NetworkInterfaceId": "eni-0065a61b324d1897a",
                    "PrivateIp": "10.0.20.33",
                    "IsPrimary": false,
```

```
                    "Status": "succeeded"
                },
                {
                    "NetworkInterfaceId": "eni-0065a61b324d1897a",
                    "PrivateIp": "10.0.20.197",
                    "IsPrimary": false,
                    "Status": "succeeded"
                }
            ],
            "NatGatewayId": "nat-1234567890abcdef0",
            "State": "available",
            "SubnetId": "subnet-08fc749671EXAMPLE",
            "VpcId": "vpc-098eb5ef58EXAMPLE",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "private-nat"
                }
            ],
            "ConnectivityType": "private"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeNatGatewaysà la section Référence des AWS CLI commandes.

`describe-network-acls`

L'exemple de code suivant montre comment utiliser`describe-network-acls`.

AWS CLI

Pour décrire votre réseau ACLs

L'`describe-network-acls`exemple suivant permet de récupérer des informations sur votre réseau ACLs.

```
aws ec2 describe-network-acls
```

Sortie :


```
{
    "NetworkAcls": [
        {
            "Associations": [
                {
                    "NetworkAclAssociationId": "aclassoc-0c1679dc41EXAMPLE",
                    "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
                    "SubnetId": "subnet-0931fc2fa5EXAMPLE"
                }
            ],
            "Entries": [
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": true,
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": true,
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32767
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32767
                }
            ],
            "IsDefault": true,
            "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
```

```
            "Tags": [],
            "VpcId": "vpc-06e4ab6c6cEXAMPLE",
            "OwnerId": "111122223333"
        },
        {
            "Associations": [],
            "Entries": [
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": true,
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "Egress": true,
                    "Ipv6CidrBlock": "::/0",
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 101
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": true,
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32767
                },
                {
                    "Egress": true,
                    "Ipv6CidrBlock": "::/0",
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32768
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "Egress": false,
```

```
                    "Ipv6CidrBlock": "::/0",
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 101
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32767
                },
                {
                    "Egress": false,
                    "Ipv6CidrBlock": "::/0",
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32768
                }
            ],
            "IsDefault": true,
            "NetworkAclId": "acl-0e2a78e4e2EXAMPLE",
            "Tags": [],
            "VpcId": "vpc-03914afb3eEXAMPLE",
            "OwnerId": "111122223333"
        }
    ]
}
```

Pour plus d'informations, consultez la section Réseau ACLs dans le guide de l'utilisateur du AWS VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkAclsà la section Référence des AWS CLI commandes.

`describe-network-insights-access-scope-analyses`

L'exemple de code suivant montre comment utiliser`describe-network-insights-access-scope-analyses`.

AWS CLI

Pour décrire les analyses du périmètre d'accès de Network Insights

L'`describe-network-insights-access-scope-analyses`exemple suivant décrit l'analyse de l'étendue d'accès de votre AWS compte.

```
aws ec2 describe-network-insights-access-scope-analyses \
    --region us-east-1
```

Sortie :

```
{
    "NetworkInsightsAccessScopeAnalyses": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111",
            "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789111",
            "NetworkInsightsAccessScopeId": "nis-123456789222",
            "Status": "succeeded",
            "StartDate": "2022-01-25T19:45:36.842000+00:00",
            "FindingsFound": "true",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInsightsAccessScopeAnalysesà la section Référence des AWS CLI commandes.

`describe-network-insights-access-scopes`

L'exemple de code suivant montre comment utiliser`describe-network-insights-access-scopes`.

AWS CLI

Pour décrire les étendues d'accès à Network Insights

L'`describe-network-insights-access-scopes`exemple suivant décrit les analyses du périmètre d'accès de votre AWS compte.

```
aws ec2 describe-network-insights-access-scopes \
```

```
    --region us-east-1
```

Sortie :

```
{
    "NetworkInsightsAccessScopes": [
        {
            "NetworkInsightsAccessScopeId": "nis-123456789111",
            "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789111",
            "CreatedDate": "2021-11-29T21:12:41.416000+00:00",
            "UpdatedDate": "2021-11-29T21:12:41.416000+00:00",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInsightsAccessScopesà la section Référence des AWS CLI commandes.

`describe-network-insights-analyses`

L'exemple de code suivant montre comment utiliser`describe-network-insights-analyses`.

AWS CLI

Pour afficher les résultats d'une analyse de trajectoire

L'`describe-network-insights-analyses`exemple suivant décrit l'analyse spécifiée. Dans cet exemple, la source est une passerelle Internet, la destination est une EC2 instance et le protocole est TCP. L'analyse a réussi (`Status`est`succeeded`) et le chemin n'est pas accessible (`NetworkPathFound`est`false`). Le code d'explication `ENI_SG_RULES_MISMATCH` indique que le groupe de sécurité de l'instance ne contient pas de règle autorisant le trafic sur le port de destination.

```
aws ec2 describe-network-insights-analyses \
    --network-insights-analysis-ids nia-02207aa13eb480c7a
```

Sortie :


```
{
    "NetworkInsightsAnalyses": [
        {
            "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
            "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-analysis/nia-02207aa13eb480c7a",
            "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
            "StartDate": "2021-01-20T22:58:37.495Z",
            "Status": "succeeded",
            "NetworkPathFound": false,
            "Explanations": [
                {
                    "Direction": "ingress",
                    "ExplanationCode": "ENI_SG_RULES_MISMATCH",
                    "NetworkInterface": {
                        "Id": "eni-0a25edef15a6cc08c",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a25edef15a6cc08c"
                    },
                    "SecurityGroups": [
                        {
                            "Id": "sg-02f0d35a850ba727f",
                            "Arn": "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-02f0d35a850ba727f"
                        }
                    ],
                    "Subnet": {
                        "Id": "subnet-004ff41eccb4d1194",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    }
                }
            ],
            "Tags": []
        }
    ]
```

```
}
```

Pour plus d'informations, consultez Getting started using the AWS CLI dans le guide Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInsightsAnalysesà la section Référence des AWS CLI commandes.

`describe-network-insights-paths`

L'exemple de code suivant montre comment utiliser`describe-network-insights-paths`.

AWS CLI

Pour décrire un chemin

L'`describe-network-insights-paths`exemple suivant décrit le chemin spécifié.

```
aws ec2 describe-network-insights-paths \
    --network-insights-path-ids nip-0b26f224f1d131fa8
```

Sortie :

```
{
    "NetworkInsightsPaths": [
        {
            "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
            "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
            "CreatedDate": "2021-01-20T22:43:46.933Z",
            "Source": "igw-0797cccdc9d73b0e5",
            "Destination": "i-0495d385ad28331c7",
            "Protocol": "tcp"
        }
    ]
}
```

Pour plus d'informations, consultez Getting started using the AWS CLI dans le guide Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInsightsPathsà la section Référence des AWS CLI commandes.

`describe-network-interface-attribute`

L'exemple de code suivant montre comment utiliser`describe-network-interface-attribute`.

AWS CLI

Pour décrire l'attribut de pièce jointe d'une interface réseau

Cet exemple de commande décrit l'`attachment`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute attachment
```

Sortie :

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Attachment": {
      "Status": "attached",
      "DeviceIndex": 0,
      "AttachTime": "2015-05-21T20:02:20.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "DeleteOnTermination": true,
      "AttachmentId": "eni-attach-43348162",
      "InstanceOwnerId": "123456789012"
  }
}
```

Pour décrire l'attribut de description d'une interface réseau

Cet exemple de commande décrit l'`description`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute description
```

Sortie :

```
{
```

```
  "NetworkInterfaceId": "eni-686ea200",
  "Description": {
      "Value": "My description"
  }
}
```

Pour décrire l'attribut GroupSet d'une interface réseau

Cet exemple de commande décrit l'`groupSet`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute groupSet
```

Sortie :

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
      {
          "GroupName": "my-security-group",
          "GroupId": "sg-903004f8"
      }
  ]
}
```

Pour décrire l' sourceDestCheck attribut d'une interface réseau

Cet exemple de commande décrit l'`sourceDestCheck`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute sourceDestCheck
```

Sortie :

```
{
  "NetworkInterfaceId": "eni-686ea200",
```

```
  "SourceDestCheck": {
      "Value": true
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInterfaceAttributeà la section Référence des AWS CLI commandes.

`describe-network-interface-permissions`

L'exemple de code suivant montre comment utiliser`describe-network-interface-permissions`.

AWS CLI

Pour décrire les autorisations de votre interface réseau

Cet exemple décrit toutes les autorisations de votre interface réseau.

Commande :

```
aws ec2 describe-network-interface-permissions
```

Sortie :

```
{
  "NetworkInterfacePermissions": [
      {
          "PermissionState": {
              "State": "GRANTED"
          },
          "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
          "NetworkInterfaceId": "eni-b909511a",
          "Permission": "INSTANCE-ATTACH",
          "AwsAccountId": "123456789012"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInterfacePermissionsà la section Référence des AWS CLI commandes.

`describe-network-interfaces`

L'exemple de code suivant montre comment utiliser`describe-network-interfaces`.

AWS CLI

Pour décrire vos interfaces réseau

Cet exemple décrit toutes vos interfaces réseau.

Commande :

```
aws ec2 describe-network-interfaces
```

Sortie :

```
{
  "NetworkInterfaces": [
      {
          "Status": "in-use",
          "MacAddress": "02:2f:8f:b0:cf:75",
          "SourceDestCheck": true,
          "VpcId": "vpc-a01106c2",
          "Description": "my network interface",
          "Association": {
              "PublicIp": "203.0.113.12",
              "AssociationId": "eipassoc-0fbb766a",
              "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
              "IpOwnerId": "123456789012"
          },
          "NetworkInterfaceId": "eni-e5aa89a3",
          "PrivateIpAddresses": [
              {
                  "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
                  "Association": {
                      "PublicIp": "203.0.113.12",
                      "AssociationId": "eipassoc-0fbb766a",
                      "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
                      "IpOwnerId": "123456789012"
                  },
                  "Primary": true,
                  "PrivateIpAddress": "10.0.1.17"
              }
          ],
```

```
            "RequesterManaged": false,
            "Ipv6Addresses": [],
            "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
            "AvailabilityZone": "us-east-1d",
            "Attachment": {
                "Status": "attached",
                "DeviceIndex": 1,
                "AttachTime": "2013-11-30T23:36:42.000Z",
                "InstanceId": "i-1234567890abcdef0",
                "DeleteOnTermination": false,
                "AttachmentId": "eni-attach-66c4350a",
                "InstanceOwnerId": "123456789012"
            },
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-8637d3e3"
                }
            ],
            "SubnetId": "subnet-b61f49f0",
            "OwnerId": "123456789012",
            "TagSet": [],
            "PrivateIpAddress": "10.0.1.17"
        },
        {
            "Status": "in-use",
            "MacAddress": "02:58:f5:ef:4b:06",
            "SourceDestCheck": true,
            "VpcId": "vpc-a01106c2",
            "Description": "Primary network interface",
            "Association": {
                "PublicIp": "198.51.100.0",
                "IpOwnerId": "amazon"
            },
            "NetworkInterfaceId": "eni-f9ba99bf",
            "PrivateIpAddresses": [
                {
                    "Association": {
                        "PublicIp": "198.51.100.0",
                        "IpOwnerId": "amazon"
                    },
                    "Primary": true,
                    "PrivateIpAddress": "10.0.1.149"
                }
```

```
            ],
            "RequesterManaged": false,
            "Ipv6Addresses": [],
            "AvailabilityZone": "us-east-1d",
            "Attachment": {
                "Status": "attached",
                "DeviceIndex": 0,
                "AttachTime": "2013-11-30T23:35:33.000Z",
                "InstanceId": "i-0598c7d356eba48d7",
                "DeleteOnTermination": true,
                "AttachmentId": "eni-attach-1b9db777",
                "InstanceOwnerId": "123456789012"
            },
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-8637d3e3"
                }
            ],
            "SubnetId": "subnet-b61f49f0",
            "OwnerId": "123456789012",
            "TagSet": [],
            "PrivateIpAddress": "10.0.1.149"
        }
    ]
}
```

Cet exemple décrit les interfaces réseau dotées d'une balise avec la clé `Purpose` et la valeur`Prod`.

Commande :

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

Sortie :

```
{
    "NetworkInterfaces": [
        {
            "Status": "available",
            "MacAddress": "12:2c:bd:f9:bf:17",
            "SourceDestCheck": true,
            "VpcId": "vpc-8941ebec",
```

```
            "Description": "ProdENI",
            "NetworkInterfaceId": "eni-b9a5ac93",
            "PrivateIpAddresses": [
                {
                    "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
                    "Primary": true,
                    "PrivateIpAddress": "10.0.1.55"
                },
                {
                    "PrivateDnsName": "ip-10-0-1-117.ec2.internal",
                    "Primary": false,
                    "PrivateIpAddress": "10.0.1.117"
                }
            ],
            "RequesterManaged": false,
            "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
            "AvailabilityZone": "us-east-1d",
            "Ipv6Addresses": [],
            "Groups": [
                {
                    "GroupName": "MySG",
                    "GroupId": "sg-905002f5"
                }
            ],
            "SubnetId": "subnet-31d6c219",
            "OwnerId": "123456789012",
            "TagSet": [
                {
                    "Value": "Prod",
                    "Key": "Purpose"
                }
            ],
            "PrivateIpAddress": "10.0.1.55"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeNetworkInterfacesà la section Référence des AWS CLI commandes.

`describe-placement-groups`

L'exemple de code suivant montre comment utiliser`describe-placement-groups`.

AWS CLI

Pour décrire vos groupes de placement

Cet exemple de commande décrit tous vos groupes de placement.

Commande :

```
aws ec2 describe-placement-groups
```

Sortie :

```
{
    "PlacementGroups": [
        {
            "GroupName": "my-cluster",
            "State": "available",
            "Strategy": "cluster"
        },
        ...
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePlacementGroupsà la section Référence des AWS CLI commandes.

`describe-prefix-lists`

L'exemple de code suivant montre comment utiliser`describe-prefix-lists`.

AWS CLI

Pour décrire les listes de préfixes

Cet exemple répertorie toutes les listes de préfixes disponibles pour la région.

Commande :

```
aws ec2 describe-prefix-lists
```

Sortie :

```
{
  "PrefixLists": [
    {
      "PrefixListName": "com.amazonaws.us-east-1.s3",
      "Cidrs": [
        "54.231.0.0/17"
      ],
      "PrefixListId": "pl-63a5400a"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePrefixListsà la section Référence des AWS CLI commandes.

`describe-principal-id-format`

L'exemple de code suivant montre comment utiliser`describe-principal-id-format`.

AWS CLI

Pour décrire le format d'identifiant pour les utilisateurs et les rôles IAM pour lesquels le format d'identifiant long est activé

L'`describe-principal-id-format`exemple suivant décrit le format d'identification de l'utilisateur root, de tous les rôles IAM et de tous les utilisateurs IAM pour lesquels le format d'identifiant long est activé.

```
aws ec2 describe-principal-id-format \
    --resource instance
```

Sortie :

```
{
    "Principals": [
        {
            "Arn": "arn:aws:iam::123456789012:root",
            "Statuses": [
                {
                    "Deadline": "2016-12-15T00:00:00.000Z",
                    "Resource": "reservation",
```

```
                    "UseLongIds": true
                },
                {
                    "Deadline": "2016-12-15T00:00:00.000Z",
                    "Resource": "instance",
                    "UseLongIds": true
                },
                {
                    "Deadline": "2016-12-15T00:00:00.000Z",
                    "Resource": "volume",
                    "UseLongIds": true
                },
            ]
        },
        ...
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePrincipalIdFormatà la section Référence des AWS CLI commandes.

`describe-public-ipv4-pools`

L'exemple de code suivant montre comment utiliser`describe-public-ipv4-pools`.

AWS CLI

Pour décrire vos pools d' IPv4 adresses publics

L'`describe-public-ipv4-pools`exemple suivant affiche des détails sur les pools d'adresses créés lorsque vous avez provisionné des plages d' IPv4 adresses publiques à l'aide de Bring Your Own IP Addresses (BYOIP).

```
aws ec2 describe-public-ipv4-pools
```

Sortie :

```
{
    "PublicIpv4Pools": [
        {
            "PoolId": "ipv4pool-ec2-1234567890abcdef0",
            "PoolAddressRanges": [
```

```
                {
                    "FirstAddress": "203.0.113.0",
                    "LastAddress": "203.0.113.255",
                    "AddressCount": 256,
                    "AvailableAddressCount": 256
                }
            ],
            "TotalAddressCount": 256,
            "TotalAvailableAddressCount": 256
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribePublicIpv4Pools dans AWS CLI Command Reference.

`describe-regions`

L'exemple de code suivant montre comment utiliser`describe-regions`.

AWS CLI

Exemple 1 : description de l'ensemble de vos régions activées

L'exemple `describe-regions` suivant décrit toutes les régions activées pour votre compte.

```
aws ec2 describe-regions
```

Sortie :

```
{
    "Regions": [
        {
            "Endpoint": "ec2.eu-north-1.amazonaws.com",
            "RegionName": "eu-north-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-south-1.amazonaws.com",
            "RegionName": "ap-south-1",
            "OptInStatus": "opt-in-not-required"
        },
```

```
        {
            "Endpoint": "ec2.eu-west-3.amazonaws.com",
            "RegionName": "eu-west-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-2.amazonaws.com",
            "RegionName": "eu-west-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-1.amazonaws.com",
            "RegionName": "eu-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
            "RegionName": "ap-northeast-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
            "RegionName": "ap-northeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
            "RegionName": "ap-northeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.sa-east-1.amazonaws.com",
            "RegionName": "sa-east-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ca-central-1.amazonaws.com",
            "RegionName": "ca-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
            "RegionName": "ap-southeast-1",
            "OptInStatus": "opt-in-not-required"
```

```
        },
        {
            "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
            "RegionName": "ap-southeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-central-1.amazonaws.com",
            "RegionName": "eu-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-2.amazonaws.com",
            "RegionName": "us-east-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-1.amazonaws.com",
            "RegionName": "us-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-2.amazonaws.com",
            "RegionName": "us-west-2",
            "OptInStatus": "opt-in-not-required"
        }
    ]
}
```

Pour plus d'informations, consultez Régions et zones dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : description des régions activées avec un point de terminaison dont le nom contient une chaîne de caractères spécifique

L'exemple `describe-regions` suivant décrit toutes les régions que vous avez activées et dont le point de terminaison contient la chaîne « us ».

```
aws ec2 describe-regions \
```

```
    --filters "Name=endpoint,Values=*us*"
```

Sortie :

```
{
    "Regions": [
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1"
        },
        {
            "Endpoint": "ec2.us-east-2.amazonaws.com",
            "RegionName": "us-east-2"
        },
        {
            "Endpoint": "ec2.us-west-1.amazonaws.com",
            "RegionName": "us-west-1"
        },
        {
            "Endpoint": "ec2.us-west-2.amazonaws.com",
            "RegionName": "us-west-2"
        }
    ]
}
```

Pour plus d'informations, consultez Régions et zones dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : description de toutes les régions

L'exemple `describe-regions` suivant décrit toutes les régions disponibles, y compris les régions désactivées.

```
aws ec2 describe-regions \
    --all-regions
```

Sortie :

```
{
    "Regions": [
        {
            "Endpoint": "ec2.eu-north-1.amazonaws.com",
            "RegionName": "eu-north-1",
            "OptInStatus": "opt-in-not-required"
```

```
        },
        {
            "Endpoint": "ec2.ap-south-1.amazonaws.com",
            "RegionName": "ap-south-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-3.amazonaws.com",
            "RegionName": "eu-west-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-2.amazonaws.com",
            "RegionName": "eu-west-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-1.amazonaws.com",
            "RegionName": "eu-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
            "RegionName": "ap-northeast-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.me-south-1.amazonaws.com",
            "RegionName": "me-south-1",
            "OptInStatus": "not-opted-in"
        },
        {
            "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
            "RegionName": "ap-northeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
            "RegionName": "ap-northeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.sa-east-1.amazonaws.com",
            "RegionName": "sa-east-1",
```

```
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ca-central-1.amazonaws.com",
            "RegionName": "ca-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-east-1.amazonaws.com",
            "RegionName": "ap-east-1",
            "OptInStatus": "not-opted-in"
        },
        {
            "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
            "RegionName": "ap-southeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
            "RegionName": "ap-southeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-central-1.amazonaws.com",
            "RegionName": "eu-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-2.amazonaws.com",
            "RegionName": "us-east-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-1.amazonaws.com",
            "RegionName": "us-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-2.amazonaws.com",
```

```
            "RegionName": "us-west-2",
            "OptInStatus": "opt-in-not-required"
        }
    ]
}
```

Pour plus d'informations, consultez Régions et zones dans le guide de EC2 l'utilisateur Amazon.

Exemple 4 : énumération des noms de région uniquement

L'exemple `describe-regions` suivant utilise le paramètre `--query` pour filtrer la sortie et ne renvoyer que les noms des régions sous forme de texte.

```
aws ec2 describe-regions \
    --all-regions \
    --query "Regions[].{Name:RegionName}" \
    --output text
```

Sortie :

```
eu-north-1
ap-south-1
eu-west-3
eu-west-2
eu-west-1
ap-northeast-3
ap-northeast-2
me-south-1
ap-northeast-1
sa-east-1
ca-central-1
ap-east-1
ap-southeast-1
ap-southeast-2
eu-central-1
us-east-1
us-east-2
us-west-1
us-west-2
```

Pour plus d'informations, consultez Régions et zones dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeRegionsà la section Référence des AWS CLI commandes.

`describe-replace-root-volume-tasks`

L'exemple de code suivant montre comment utiliser`describe-replace-root-volume-tasks`.

AWS CLI

Exemple 1 : pour afficher les informations relatives à une tâche de remplacement de volume racine spécifique

L'`describe-replace-root-volume-tasks`exemple suivant décrit la tâche de remplacement du volume racine replacevol-0111122223333abcd.

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-0111122223333abcd
```

Sortie :

```
{
    "ReplaceRootVolumeTasks": [
        {
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:16:28Z",
            "CompleteTime": "2022-03-14T15:16:52Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Remplacer un volume racine dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

Exemple 2 : pour afficher les informations relatives à toutes les tâches de remplacement du volume racine pour une instance spécifique

L'`describe-replace-root-volume-tasks`exemple suivant décrit toutes les tâches de remplacement du volume racine, par exemple i-0123456789abcdefa.

```
aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-0123456789abcdefa
```

Sortie :

```
{
    "ReplaceRootVolumeTasks": [
        {
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:06:38Z",
            "CompleteTime": "2022-03-14T15:07:03Z"
        },
        {
            "ReplaceRootVolumeTaskId": "replacevol-0444455555555abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:16:28Z",
            "CompleteTime": "2022-03-14T15:16:52Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Remplacer un volume racine dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous DescribeReplaceRootVolumeTasksà la section Référence des AWS CLI commandes.

`describe-reserved-instances-listings`

L'exemple de code suivant montre comment utiliser`describe-reserved-instances-listings`.

AWS CLI

Pour décrire une liste d'instances réservées

L'`describe-reserved-instances-listings`exemple suivant permet de récupérer des informations sur la liste des instances réservées spécifiée.

```
aws ec2 describe-reserved-instances-listings \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DescribeReservedInstancesListingsà la section Référence des AWS CLI commandes.

`describe-reserved-instances-modifications`

L'exemple de code suivant montre comment utiliser`describe-reserved-instances-modifications`.

AWS CLI

Pour décrire les modifications apportées aux instances réservées

Cet exemple de commande décrit toutes les demandes de modification des instances réservées qui ont été soumises pour votre compte.

Commande :

```
aws ec2 describe-reserved-instances-modifications
```

Sortie :

```
{
    "ReservedInstancesModifications": [
        {
            "Status": "fulfilled",
            "ModificationResults": [
                {
                    "ReservedInstancesId": "93bbbca2-62f1-4d9d-b225-16bada29e6c7",
                    "TargetConfiguration": {
                        "AvailabilityZone": "us-east-1b",
                        "InstanceType": "m1.large",
                        "InstanceCount": 3
                    }
                },
                {
                     "ReservedInstancesId": "1ba8e2e3-aabb-46c3-bcf5-3fe2fda922e6",
```

```
                    "TargetConfiguration": {
                        "AvailabilityZone": "us-east-1d",
                        "InstanceType": "m1.xlarge",
                        "InstanceCount": 1
                    }
                }
            ],
            "EffectiveDate": "2015-08-12T17:00:00.000Z",
            "CreateDate": "2015-08-12T17:52:52.630Z",
            "UpdateDate": "2015-08-12T18:08:06.698Z",
            "ClientToken": "c9adb218-3222-4889-8216-0cf0e52dc37e:
            "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-
ab31-0f13aaf46687",
            "ReservedInstancesIds": [
                {
                    "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342f5bd7c02"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeReservedInstancesModificationsà la section Référence des AWS CLI commandes.

`describe-reserved-instances-offerings`

L'exemple de code suivant montre comment utiliser`describe-reserved-instances-offerings`.

AWS CLI

Pour décrire les offres d'instances réservées

Cet exemple de commande décrit toutes les instances réservées disponibles à l'achat dans la région.

Commande :

```
aws ec2 describe-reserved-instances-offerings
```

Sortie :

```
{
  "ReservedInstancesOfferings": [
      {
          "OfferingType": "Partial Upfront",
          "AvailabilityZone": "us-east-1b",
          "InstanceTenancy": "default",
          "PricingDetails": [],
          "ProductDescription": "Red Hat Enterprise Linux",
          "UsagePrice": 0.0,
          "RecurringCharges": [
              {
                  "Amount": 0.088,
                  "Frequency": "Hourly"
              }
          ],
          "Marketplace": false,
          "CurrencyCode": "USD",
          "FixedPrice": 631.0,
          "Duration": 94608000,
          "ReservedInstancesOfferingId": "9a06095a-bdc6-47fe-a94a-2a382f016040",
          "InstanceType": "c1.medium"
      },
      {
          "OfferingType": "PartialUpfront",
          "AvailabilityZone": "us-east-1b",
          "InstanceTenancy": "default",
          "PricingDetails": [],
          "ProductDescription": "Linux/UNIX",
          "UsagePrice": 0.0,
          "RecurringCharges": [
              {
                  "Amount": 0.028,
                  "Frequency": "Hourly"
              }
          ],
          "Marketplace": false,
          "CurrencyCode": "USD",
          "FixedPrice": 631.0,
          "Duration": 94608000,
          "ReservedInstancesOfferingId": "bfbefc6c-0d10-418d-b144-7258578d329d",
          "InstanceType": "c1.medium"
      },
  ...
```

```
}
```

Pour décrire vos offres d'instances réservées à l'aide d'options

Cet exemple répertorie les instances réservées proposées par AWS avec les spécifications suivantes : types d'instances t1.micro, produits Windows (Amazon VPC) et offres d'utilisation intensive.

Commande :

```
aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-
type "t1.micro" --product-description "Windows (Amazon VPC)" --offering-type "no
 upfront"
```

Sortie :

```
{
    "ReservedInstancesOfferings": [
      {
          "OfferingType": "No Upfront",
          "AvailabilityZone": "us-east-1b",
          "InstanceTenancy": "default",
          "PricingDetails": [],
          "ProductDescription": "Windows",
          "UsagePrice": 0.0,
          "RecurringCharges": [
              {
                  "Amount": 0.015,
                  "Frequency": "Hourly"
              }
          ],
          "Marketplace": false,
          "CurrencyCode": "USD",
          "FixedPrice": 0.0,
          "Duration": 31536000,
          "ReservedInstancesOfferingId": "c48ab04c-fe69-4f94-8e39-a23842292823",
          "InstanceType": "t1.micro"
      },

              ...
      {
          "OfferingType": "No Upfront",
          "AvailabilityZone": "us-east-1d",
```

```
            "InstanceTenancy": "default",
            "PricingDetails": [],
            "ProductDescription": "Windows (Amazon VPC)",
            "UsagePrice": 0.0,
            "RecurringCharges": [
                {
                    "Amount": 0.015,
                    "Frequency": "Hourly"
                }
            ],
            "Marketplace": false,
            "CurrencyCode": "USD",
            "FixedPrice": 0.0,
            "Duration": 31536000,
            "ReservedInstancesOfferingId": "3a98bf7d-2123-42d4-b4f5-8dbec4b06dc6",
            "InstanceType": "t1.micro"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeReservedInstancesOfferingsà la section Référence des AWS CLI commandes.

`describe-reserved-instances`

L'exemple de code suivant montre comment utiliser`describe-reserved-instances`.

AWS CLI

Pour décrire vos instances réservées

Cet exemple de commande décrit les instances réservées que vous possédez.

Commande :

```
aws ec2 describe-reserved-instances
```

Sortie :

```
{
  "ReservedInstances": [
      {
          "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342fexample",
```

```
            "OfferingType": "No Upfront",
            "AvailabilityZone": "us-west-1c",
            "End": "2016-08-14T21:34:34.000Z",
            "ProductDescription": "Linux/UNIX",
            "UsagePrice": 0.00,
            "RecurringCharges": [
                {
                    "Amount": 0.104,
                    "Frequency": "Hourly"
                }
            ],
            "Start": "2015-08-15T21:34:35.086Z",
            "State": "active",
            "FixedPrice": 0.0,
            "CurrencyCode": "USD",
            "Duration": 31536000,
            "InstanceTenancy": "default",
            "InstanceType": "m3.medium",
            "InstanceCount": 2
        },
        ...
    ]
}
```

Pour décrire vos instances réservées à l'aide de filtres

Cet exemple filtre la réponse pour inclure uniquement les instances Linux/UNIX réservées t2.micro de trois ans dans us-west-1c.

Commande :

```
aws ec2 describe-reserved-instances --
filters Name=duration,Values=94608000 Name=instance-
type,Values=t2.micro Name=product-description,Values=Linux/UNIX Name=availability-
zone,Values=us-east-1e
```

Sortie :

```
{
    "ReservedInstances": [
        {
            "ReservedInstancesId": "f127bd27-edb7-44c9-a0eb-0d7e09259af0",
            "OfferingType": "All Upfront",
```

```
            "AvailabilityZone": "us-east-1e",
            "End": "2018-03-26T21:34:34.000Z",
            "ProductDescription": "Linux/UNIX",
            "UsagePrice": 0.00,
            "RecurringCharges": [],
            "Start": "2015-03-27T21:34:35.848Z",
            "State": "active",
            "FixedPrice": 151.0,
            "CurrencyCode": "USD",
            "Duration": 94608000,
            "InstanceTenancy": "default",
            "InstanceType": "t2.micro",
            "InstanceCount": 1
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des EC2 instances Amazon dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous DescribeReservedInstancesà la section Référence des AWS CLI commandes.

`describe-route-tables`

L'exemple de code suivant montre comment utiliser`describe-route-tables`.

AWS CLI

Pour décrire vos tables de routage

L'`describe-route-tables`exemple suivant permet de récupérer les détails de vos tables de routage.

```
aws ec2 describe-route-tables
```

Sortie :

```
{
    "RouteTables": [
        {
            "Associations": [
                {
```

```
                    "Main": true,
                    "RouteTableAssociationId": "rtbassoc-0df3f54e06EXAMPLE",
                    "RouteTableId": "rtb-09ba434c1bEXAMPLE"
                }
            ],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-09ba434c1bEXAMPLE",
            "Routes": [
                {
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active"
                },
                {
                    "DestinationCidrBlock": "0.0.0.0/0",
                    "NatGatewayId": "nat-06c018cbd8EXAMPLE",
                    "Origin": "CreateRoute",
                    "State": "blackhole"
                }
            ],
            "Tags": [],
            "VpcId": "vpc-0065acced4EXAMPLE",
            "OwnerId": "111122223333"
        },
        {
            "Associations": [
                {
                    "Main": true,
                    "RouteTableAssociationId": "rtbassoc-9EXAMPLE",
                    "RouteTableId": "rtb-a1eec7de"
                }
            ],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-a1eec7de",
            "Routes": [
                {
                    "DestinationCidrBlock": "172.31.0.0/16",
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active"
                },
                {
                    "DestinationCidrBlock": "0.0.0.0/0",
```

```
                    "GatewayId": "igw-fEXAMPLE",
                    "Origin": "CreateRoute",
                    "State": "active"
                }
            ],
            "Tags": [],
            "VpcId": "vpc-3EXAMPLE",
            "OwnerId": "111122223333"
        },
        {
            "Associations": [
                {
                    "Main": false,
                    "RouteTableAssociationId": "rtbassoc-0b100c28b2EXAMPLE",
                    "RouteTableId": "rtb-07a98f76e5EXAMPLE",
                    "SubnetId": "subnet-0d3d002af8EXAMPLE"
                }
            ],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-07a98f76e5EXAMPLE",
            "Routes": [
                {
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active"
                },
                {
                    "DestinationCidrBlock": "0.0.0.0/0",
                    "GatewayId": "igw-06cf664d80EXAMPLE",
                    "Origin": "CreateRoute",
                    "State": "active"
                }
            ],
            "Tags": [],
            "VpcId": "vpc-0065acced4EXAMPLE",
            "OwnerId": "111122223333"
        }
    ]
}
```


Pour plus d'informations, consultez la section Utilisation des tables de routage dans le guide de l'utilisateur AWS VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeRouteTablesà la section Référence des AWS CLI commandes.

`describe-scheduled-instance-availability`

L'exemple de code suivant montre comment utiliser`describe-scheduled-instance-availability`.

AWS CLI

Pour décrire un calendrier disponible

Cet exemple décrit un calendrier qui a lieu chaque semaine le dimanche, à compter de la date spécifiée.

Commande :

```
aws ec2 describe-scheduled-instance-availability --
recurrence Frequency=Weekly,Interval=1,OccurrenceDays=[1] --first-slot-start-time-
range EarliestTime=2016-01-31T00:00:00Z,LatestTime=2016-01-31T04:00:00Z
```

Sortie :

```
{
  "ScheduledInstanceAvailabilitySet": [
    {
        "AvailabilityZone": "us-west-2b",
        "TotalScheduledInstanceHours": 1219,
        "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
        "MinTermDurationInDays": 366,
        "AvailableInstanceCount": 20,
        "Recurrence": {
            "OccurrenceDaySet": [
                1
            ],
            "Interval": 1,
            "Frequency": "Weekly",
            "OccurrenceRelativeToEnd": false
        },
        "Platform": "Linux/UNIX",
        "FirstSlotStartTime": "2016-01-31T00:00:00Z",
```

```
        "MaxTermDurationInDays": 366,
        "SlotDurationInHours": 23,
        "NetworkPlatform": "EC2-VPC",
        "InstanceType": "c4.large",
        "HourlyPrice": "0.095"
    },
    ...
  ]
}
```

Pour affiner les résultats, vous pouvez ajouter des filtres qui spécifient le système d'exploitation, le réseau et le type d'instance.

Commande :

--filters Nom = plate-forme, valeurs = nom Linux/UNIX = plate-forme réseau, valeurs = -nom VPC = type d'instance, valeurs = C4.large EC2

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledInstanceAvailabilityà la section Référence des AWS CLI commandes.

`describe-scheduled-instances`

L'exemple de code suivant montre comment utiliser`describe-scheduled-instances`.

AWS CLI

Pour décrire vos instances planifiées

Cet exemple décrit l'instance planifiée spécifiée.

Commande :

```
aws ec2 describe-scheduled-instances --scheduled-instance-
ids sci-1234-1234-1234-1234-123456789012
```

Sortie :

```
{
  "ScheduledInstanceSet": [
      {
```

```
            "AvailabilityZone": "us-west-2b",
            "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
            "HourlyPrice": "0.095",
            "CreateDate": "2016-01-25T21:43:38.612Z",
            "Recurrence": {
                "OccurrenceDaySet": [
                    1
                ],
                "Interval": 1,
                "Frequency": "Weekly",
                "OccurrenceRelativeToEnd": false,
                "OccurrenceUnit": ""
            },
            "Platform": "Linux/UNIX",
            "TermEndDate": "2017-01-31T09:00:00Z",
            "InstanceCount": 1,
            "SlotDurationInHours": 32,
            "TermStartDate": "2016-01-31T09:00:00Z",
            "NetworkPlatform": "EC2-VPC",
            "TotalScheduledInstanceHours": 1696,
            "NextSlotStartTime": "2016-01-31T09:00:00Z",
            "InstanceType": "c4.large"
        }
    ]
}
```

Cet exemple décrit toutes vos instances planifiées.

Commande :

```
aws ec2 describe-scheduled-instances
```

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledInstancesà la section Référence des AWS CLI commandes.

`describe-security-group-references`

L'exemple de code suivant montre comment utiliser`describe-security-group-references`.

AWS CLI

Pour décrire les références aux groupes de sécurité

Cet exemple décrit les références des groupes de sécurité pour`sg-bbbb2222`. La réponse indique que le groupe de sécurité `sg-bbbb2222` est référencé par un groupe de sécurité dans le VPC`vpc-aaaaaaaa`.

Commande :

```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

Sortie :

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSecurityGroupReferencesà la section Référence des AWS CLI commandes.

`describe-security-group-rules`

L'exemple de code suivant montre comment utiliser`describe-security-group-rules`.

AWS CLI

Exemple 1 : pour décrire les règles d'un groupe de sécurité

L'`describe-security-group-rules`exemple suivant décrit les règles du groupe de sécurité d'un groupe de sécurité spécifié. Utilisez `filters` cette option pour étendre les résultats à un groupe de sécurité spécifique.

```
aws ec2 describe-security-group-rules \
    --filters Name="group-id",Values="sg-1234567890abcdef0"
```

Sortie :

```
{
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-abcdef01234567890",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": false,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "ReferencedGroupInfo": {
                "GroupId": "sg-1234567890abcdef0",
                "UserId": "111122223333"
            },
            "Tags": []
        },
        {
            "SecurityGroupRuleId": "sgr-bcdef01234567890a",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv6": "::/0",
            "Tags": []
        },
        {
            "SecurityGroupRuleId": "sgr-cdef01234567890ab",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "0.0.0.0/0",
            "Tags": []
        }
    ]
}
```


Exemple 2 : pour décrire une règle de groupe de sécurité

L'`describe-security-group-rules`exemple suivant décrit la règle de groupe de sécurité spécifiée.

```
aws ec2 describe-security-group-rules \
    --security-group-rule-ids sgr-cdef01234567890ab
```

Sortie :

```
{
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-cdef01234567890ab",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "0.0.0.0/0",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez Règles des groupes de sécurité dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeSecurityGroupRulesà la section Référence des AWS CLI commandes.

`describe-security-group-vpc-associations`

L'exemple de code suivant montre comment utiliser`describe-security-group-vpc-associations`.

AWS CLI

Pour décrire les associations VPC

L'`describe-security-group-vpc-associations`exemple suivant décrit les associations VPC pour le groupe de sécurité spécifié.

```
aws ec2 describe-security-group-vpc-associations \
    --filters Name=group-id,Values=sg-04dbb43907d3f8a78
```

Sortie :

```
{
    "SecurityGroupVpcAssociations": [
        {
            "GroupId": "sg-04dbb43907d3f8a78",
            "VpcId": "vpc-0bf4c2739bc05a694",
            "VpcOwnerId": "123456789012",
            "State": "associated"
        }
    ]
}
```

Pour plus d'informations, consultez Associer des groupes de sécurité à plusieurs VPCs dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeSecurityGroupVpcAssociationsà la section Référence des AWS CLI commandes.

`describe-security-groups`

L'exemple de code suivant montre comment utiliser`describe-security-groups`.

AWS CLI

Exemple 1 : description d'un groupe de sécurité

L'exemple `describe-security-groups` suivant décrit le groupe de sécurité spécifié.

```
aws ec2 describe-security-groups \
    --group-ids sg-903004f8
```

Sortie :

```
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
```

```
                "IpProtocol": "-1",
                "IpRanges": [
                    {
                        "CidrIp": "0.0.0.0/0"
                    }
                ],
                "UserIdGroupPairs": [],
                "PrefixListIds": []
            }
        ],
        "Description": "My security group",
        "Tags": [
            {
                "Value": "SG1",
                "Key": "Name"
            }
        ],
        "IpPermissions": [
            {
                "IpProtocol": "-1",
                "IpRanges": [],
                "UserIdGroupPairs": [
                    {
                        "UserId": "123456789012",
                        "GroupId": "sg-903004f8"
                    }
                ],
                "PrefixListIds": []
            },
            {
                "PrefixListIds": [],
                "FromPort": 22,
                "IpRanges": [
                    {
                        "Description": "Access from NY office",
                        "CidrIp": "203.0.113.0/24"
                    }
                ],
                "ToPort": 22,
                "IpProtocol": "tcp",
                "UserIdGroupPairs": []
                }
        ],
        "GroupName": "MySecurityGroup",
```

```
            "VpcId": "vpc-1a2b3c4d",
            "OwnerId": "123456789012",
            "GroupId": "sg-903004f8",
        }
    ]
}
```

Exemple 2 : description des groupes de sécurité comportant des règles spécifiques

L'`describe-security-groups`exemple suivant utilise des filtres pour étendre les résultats aux groupes de sécurité dotés d'une règle autorisant le trafic SSH (port 22) et d'une règle autorisant le trafic provenant de toutes les adresses (`0.0.0.0/0`). L'exemple utilise le paramètre `--query` pour n'afficher que les noms des groupes de sécurité. Les groupes de sécurité doivent correspondre à tous les filtres pour être retournés dans les résultats ; toutefois, une règle unique ne doit pas nécessairement correspondre à tous les filtres. Par exemple, la sortie renvoie un groupe de sécurité avec une règle qui autorise le trafic SSH depuis une adresse IP spécifique et une autre règle qui autorise le trafic HTTP depuis toutes les adresses.

```
aws ec2 describe-security-groups \
    --filters Name=ip-permission.from-port,Values=22 Name=ip-permission.to-
port,Values=22 Name=ip-permission.cidr,Values='0.0.0.0/0' \
    --query "SecurityGroups[*].[GroupName]" \
    --output text
```

Sortie :

```
default
my-security-group
web-servers
launch-wizard-1
```

Exemple 3 : description des groupes de sécurité sur la base de balises

L'exemple `describe-security-groups` suivant utilise des filtres pour limiter les résultats aux groupes de sécurité incluant `test` dans le nom du groupe de sécurité et comportant la balise `Test=To-delete`. L'exemple utilise le `--query` paramètre pour afficher uniquement les noms et les groupes IDs de sécurité.

```
aws ec2 describe-security-groups \
    --filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \
```

```
    --query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

Sortie :

```
[
    {
        "Name": "testfornewinstance",
        "ID": "sg-33bb22aa"
    },
    {
        "Name": "newgrouptest",
        "ID": "sg-1a2b3c4d"
    }
]
```

Pour d'autres exemples d'utilisation de filtres de balises, consultez la section Utilisation des balises dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeSecurityGroupsà la section Référence des AWS CLI commandes.

`describe-snapshot-attribute`

L'exemple de code suivant montre comment utiliser`describe-snapshot-attribute`.

AWS CLI

Pour décrire les attributs d'un instantané

L'`describe-snapshot-attribute`exemple suivant répertorie les comptes avec lesquels un instantané est partagé.

```
aws ec2 describe-snapshot-attribute \
    --snapshot-id snap-01234567890abcedf \
    --attribute createVolumePermission
```

Sortie :

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "CreateVolumePermissions": [
```

```
        {
            "UserId": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez Partager un instantané Amazon EBS dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous DescribeSnapshotAttributeà la section Référence des AWS CLI commandes.

`describe-snapshot-tier-status`

L'exemple de code suivant montre comment utiliser`describe-snapshot-tier-status`.

AWS CLI

Pour afficher les informations d'archivage relatives à un instantané archivé

L'`describe-snapshot-tier-status`exemple suivant fournit des informations d'archivage concernant un instantané archivé.

```
aws ec2 describe-snapshot-tier-status \
    --filters "Name=snapshot-id, Values=snap-01234567890abcedf"
```

Sortie :

```
{
    "SnapshotTierStatuses": [
        {
            "Status": "completed",
            "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
            "LastTieringProgress": 100,
            "Tags": [],
            "VolumeId": "vol-01234567890abcedf",
            "LastTieringOperationState": "archival-completed",
            "StorageTier": "archive",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890abcedf",
            "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
        }
```

```
    ]
}
```

Pour plus d'informations, veuillez consulter la rubrique Afficher les instantanés archivés dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous DescribeSnapshotTierStatusà la section Référence des AWS CLI commandes.

`describe-snapshots`

L'exemple de code suivant montre comment utiliser`describe-snapshots`.

AWS CLI

Exemple 1 : description d'un instantané

L'exemple `describe-snapshots` suivant décrit l'instantané spécifié.

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-1234567890abcdef0
```

Sortie :

```
{
    "Snapshots": [
        {
            "Description": "This is my snapshot",
            "Encrypted": false,
            "VolumeId": "vol-049df61146c4d7901",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2019-02-28T21:28:32.000Z",
            "Progress": "100%",
            "OwnerId": "012345678910",
            "SnapshotId": "snap-01234567890abcdef",
            "Tags": [
                {
                    "Key": "Stack",
                    "Value": "test"
                }
            ]
```

```
        }
    ]
}
```

Pour plus d'informations, consultez les instantanés Amazon EBS dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : description d'instantanés sur la base de filtres

L'`describe-snapshots`exemple suivant utilise des filtres pour étendre les résultats aux instantanés appartenant à votre AWS compte qui se trouvent dans l'`pending`État. L'exemple utilise le `--query` paramètre pour afficher uniquement le cliché IDs et l'heure à laquelle il a été démarré.

```
aws ec2 describe-snapshots \
    --owner-ids self \
    --filters Name=status,Values=pending \
    --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"
```

Sortie :

```
[
    {
        "ID": "snap-1234567890abcdef0",
        "Time": "2019-08-04T12:48:18.000Z"
    },
    {
        "ID": "snap-066877671789bd71b",
        "Time": "2019-08-04T02:45:16.000Z
    },
    ...
]
```

L'exemple `describe-snapshots` suivant utilise des filtres pour limiter les résultats aux instantanés créés à partir du volume spécifié. L'exemple utilise le `--query` paramètre pour afficher uniquement le cliché IDs.

```
aws ec2 describe-snapshots \
    --filters Name=volume-id,Values=049df61146c4d7901 \
    --query "Snapshots[*].[SnapshotId]" \
```

```
    --output text
```

Sortie :

```
snap-1234567890abcdef0
snap-08637175a712c3fb9
...
```

Pour d'autres exemples d'utilisation de filtres, consultez la section Répertorier et filtrer vos ressources dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : description d'instantanés sur la base de balises

L'exemple `describe-snapshots` suivant utilise des filtres de balises pour limiter les résultats aux instantanés comportant la balise `Stack=Prod`.

```
aws ec2 describe-snapshots \
    --filters Name=tag:Stack,Values=prod
```

Pour obtenir un exemple de la manière de procéder pour `describe-snapshots`, reportez-vous à l'exemple 1.

Pour d'autres exemples d'utilisation de filtres de balises, consultez la section Utilisation des balises dans le guide de EC2 l'utilisateur Amazon.

Exemple 4 : description d'instantanés en fonction de l'âge

L'`describe-snapshots`exemple suivant utilise des JMESPath expressions pour décrire tous les instantanés créés par votre AWS compte avant la date spécifiée. Il affiche uniquement l'instantané IDs.

```
aws ec2 describe-snapshots \
    --owner-ids 012345678910 \
    --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

Pour d'autres exemples d'utilisation de filtres, consultez la section Répertorier et filtrer vos ressources dans le guide de EC2 l'utilisateur Amazon.

Exemple 5 : affichage des instantanés archivés uniquement

L'exemple `describe-snapshots` suivant ne liste que les instantanés qui sont stockés dans le niveau d'archivage spécifié.

```
aws ec2 describe-snapshots \
    --filters "Name=storage-tier,Values=archive"
```

Sortie :

```
{
    "Snapshots": [
        {
            "Description": "Snap A",
            "Encrypted": false,
            "VolumeId": "vol-01234567890aaaaaa",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-09-07T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890aaaaaa",
            "StorageTier": "archive",
            "Tags": []
        },
    ]
}
```

Pour plus d'informations, veuillez consulter la rubrique Afficher les instantanés archivés dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous DescribeSnapshotsà la section Référence des AWS CLI commandes.

`describe-spot-datafeed-subscription`

L'exemple de code suivant montre comment utiliser`describe-spot-datafeed-subscription`.

AWS CLI

Pour décrire l'abonnement au flux de données d'une instance Spot pour un compte

Cet exemple de commande décrit le flux de données du compte.

Commande :

```
aws ec2 describe-spot-datafeed-subscription
```

Sortie :

```
{
    "SpotDatafeedSubscription": {
        "OwnerId": "123456789012",
        "Prefix": "spotdata",
        "Bucket": "amzn-s3-demo-bucket",
        "State": "Active"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSpotDatafeedSubscriptionà la section Référence des AWS CLI commandes.

`describe-spot-fleet-instances`

L'exemple de code suivant montre comment utiliser`describe-spot-fleet-instances`.

AWS CLI

Pour décrire les instances Spot associées à un parc Spot

Cet exemple de commande répertorie les instances Spot associées au parc Spot spécifié.

Commande :

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE
```

Sortie :

```
{
  "ActiveInstances": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "InstanceType": "m3.medium",
          "SpotInstanceRequestId": "sir-08b93456"
      },
```

```
        ...
    ],
    "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSpotFleetInstancesà la section Référence des AWS CLI commandes.

`describe-spot-fleet-request-history`

L'exemple de code suivant montre comment utiliser`describe-spot-fleet-request-history`.

AWS CLI

Pour décrire l'historique de la flotte Spot

Cet exemple de commande renvoie l'historique du parc de Spot spécifié à partir de l'heure spécifiée.

Commande :

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

L'exemple de sortie suivant montre les lancements réussis de deux instances Spot pour le parc Spot.

Sortie :

```
{
    "HistoryRecords": [
        {
            "Timestamp": "2015-05-26T23:17:20.697Z",
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange"
        },
        {
            "Timestamp": "2015-05-26T23:17:20.873Z",
            "EventInformation": {
                "EventSubType": "active"
```

```
            },
            "EventType": "fleetRequestChange"
        },
        {
            "Timestamp": "2015-05-26T23:21:21.712Z",
            "EventInformation": {
                "InstanceId": "i-1234567890abcdef0",
                "EventSubType": "launched"
            },
            "EventType": "instanceChange"
        },
        {
            "Timestamp": "2015-05-26T23:21:21.816Z",
            "EventInformation": {
                "InstanceId": "i-1234567890abcdef1",
                "EventSubType": "launched"
            },
            "EventType": "instanceChange"
        }
    ],
    "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "NextToken": "CpHNsscimcV5oH7bSbub03CI2Qms5+ypNpNm
+53MNlR0YcXAkp0xFlfKf91yVxSExmbtma3awYxMFzNA663ZskT0AHtJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauVg
+P+TfGlWxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
    "StartTime": "2015-05-26T00:00:00Z"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSpotFleetRequestHistoryà la section Référence des AWS CLI commandes.

`describe-spot-fleet-requests`

L'exemple de code suivant montre comment utiliser`describe-spot-fleet-requests`.

AWS CLI

Pour décrire vos demandes relatives à une flotte Spot

Cet exemple décrit toutes vos demandes de flotte Spot.

Commande :

```
aws ec2 describe-spot-fleet-requests
```

Sortie :


```
{
  "SpotFleetRequestConfigs": [
      {
          "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
          "SpotFleetRequestConfig": {
              "TargetCapacity": 20,
              "LaunchSpecifications": [
                  {
                      "EbsOptimized": false,
                      "NetworkInterfaces": [
                          {
                              "SubnetId": "subnet-a61dafcf",
                              "DeviceIndex": 0,
                              "DeleteOnTermination": false,
                              "AssociatePublicIpAddress": true,
                              "SecondaryPrivateIpAddressCount": 0
                          }
                      ],
                      "InstanceType": "cc2.8xlarge",
                      "ImageId": "ami-1a2b3c4d"
                  },
                  {
                      "EbsOptimized": false,
                      "NetworkInterfaces": [
                          {
                              "SubnetId": "subnet-a61dafcf",
                              "DeviceIndex": 0,
                              "DeleteOnTermination": false,
                              "AssociatePublicIpAddress": true,
                              "SecondaryPrivateIpAddressCount": 0
                          }
                      ],
                      "InstanceType": "r3.8xlarge",
                      "ImageId": "ami-1a2b3c4d"
                  }
              ],
              "SpotPrice": "0.05",
              "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
          },
          "SpotFleetRequestState": "active"
      },
      {
```

```
            "SpotFleetRequestId": "sfr-306341ed-9739-402e-881b-ce47bEXAMPLE",
            "SpotFleetRequestConfig": {
                "TargetCapacity": 20,
                "LaunchSpecifications": [
                    {
                        "EbsOptimized": false,
                        "NetworkInterfaces": [
                            {
                                "SubnetId": "subnet-6e7f829e",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": false,
                                "AssociatePublicIpAddress": true,
                                "SecondaryPrivateIpAddressCount": 0
                            }
                        ],
                        "InstanceType": "m3.medium",
                        "ImageId": "ami-1a2b3c4d"
                    }
                ],
                "SpotPrice": "0.05",
                "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
            },
            "SpotFleetRequestState": "active"
        }
    ]
}
```

Pour décrire une demande de flotte Spot

Cet exemple décrit la demande de parc Spot spécifiée.

Commande :

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Sortie :

```
{
    "SpotFleetRequestConfigs": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
```

```
            "SpotFleetRequestConfig": {
                "TargetCapacity": 20,
                "LaunchSpecifications": [
                    {
                        "EbsOptimized": false,
                        "NetworkInterfaces": [
                            {
                                "SubnetId": "subnet-a61dafcf",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": false,
                                "AssociatePublicIpAddress": true,
                                "SecondaryPrivateIpAddressCount": 0
                            }
                        ],
                        "InstanceType": "cc2.8xlarge",
                        "ImageId": "ami-1a2b3c4d"
                    },
                    {
                        "EbsOptimized": false,
                        "NetworkInterfaces": [
                            {
                                "SubnetId": "subnet-a61dafcf",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": false,
                                "AssociatePublicIpAddress": true,
                                "SecondaryPrivateIpAddressCount": 0
                            }
                        ],
                        "InstanceType": "r3.8xlarge",
                        "ImageId": "ami-1a2b3c4d"
                    }
                ],
                "SpotPrice": "0.05",
                "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
            },
            "SpotFleetRequestState": "active"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSpotFleetRequestsà la section Référence des AWS CLI commandes.

`describe-spot-instance-requests`

L'exemple de code suivant montre comment utiliser`describe-spot-instance-requests`.

AWS CLI

Exemple 1 : Pour décrire une demande d'instance Spot

L'`describe-spot-instance-requests`exemple suivant décrit la demande d'instance Spot spécifiée.

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-08b93456
```

Sortie :

```
{
    "SpotInstanceRequests": [
        {
            "CreateTime": "2018-04-30T18:14:55.000Z",
            "InstanceId": "i-1234567890abcdef1",
            "LaunchSpecification": {
                "InstanceType": "t2.micro",
                "ImageId": "ami-003634241a8fcdec0",
                "KeyName": "my-key-pair",
                "SecurityGroups": [
                    {
                        "GroupName": "default",
                        "GroupId": "sg-e38f24a7"
                    }
                ],
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "DeleteOnTermination": true,
                            "SnapshotId": "snap-0e54a519c999adbbd",
                            "VolumeSize": 8,
                            "VolumeType": "standard",
                            "Encrypted": false
                        }
                    }
                ],
```

```
                "NetworkInterfaces": [
                    {
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "SubnetId": "subnet-049df61146c4d7901"
                    }
                ],
                "Placement": {
                    "AvailabilityZone": "us-east-2b",
                    "Tenancy": "default"
                },
                "Monitoring": {
                    "Enabled": false
                }
            },
            "LaunchedAvailabilityZone": "us-east-2b",
            "ProductDescription": "Linux/UNIX",
            "SpotInstanceRequestId": "sir-08b93456",
            "SpotPrice": "0.010000"
            "State": "active",
            "Status": {
                "Code": "fulfilled",
                "Message": "Your Spot request is fulfilled.",
                "UpdateTime": "2018-04-30T18:16:21.000Z"
            },
            "Tags": [],
            "Type": "one-time",
            "InstanceInterruptionBehavior": "terminate"
        }
    ]
}
```

Exemple 2 : pour décrire les demandes d'instance Spot basées sur des filtres

L'`describe-spot-instance-requests`exemple suivant utilise des filtres pour étendre les résultats aux demandes d'instance Spot avec le type d'instance spécifié dans la zone de disponibilité spécifiée. L'exemple utilise le `--query` paramètre pour afficher uniquement l'instance IDs.

```
aws ec2 describe-spot-instance-requests \
    --filters Name=launch.instance-type,Values=m3.medium Name=launched-availability-zone,Values=us-east-2a \
    --query "SpotInstanceRequests[*].[InstanceId]" \
```

```
    --output text
```

Sortie :

```
i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...
```

Pour d'autres exemples d'utilisation de filtres, consultez la section Répertorier et filtrer vos ressources dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

Exemple 3 : pour décrire les demandes d'instance Spot basées sur des balises

L'`describe-spot-instance-requests`exemple suivant utilise des filtres de balises pour étendre les résultats aux demandes d'instance Spot dotées de la balise`cost-center=cc123`.

```
aws ec2 describe-spot-instance-requests \
    --filters Name=tag:cost-center,Values=cc123
```

Pour obtenir un exemple de la manière de procéder pour `describe-spot-instance-requests`, reportez-vous à l'exemple 1.

Pour d'autres exemples d'utilisation de filtres de balises, consultez la section Utilisation des balises dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeSpotInstanceRequestsà la section Référence des AWS CLI commandes.

`describe-spot-price-history`

L'exemple de code suivant montre comment utiliser`describe-spot-price-history`.

AWS CLI

Pour décrire l'historique des prix au comptant

Cet exemple de commande renvoie l'historique des prix au comptant pour les instances m1.xlarge pour un jour donné de janvier.

Commande :

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-
time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

Sortie :

```
{
  "SpotPriceHistory": [
          {
              "Timestamp": "2014-01-06T07:10:55.000Z",
              "ProductDescription": "SUSE Linux",
              "InstanceType": "m1.xlarge",
              "SpotPrice": "0.087000",
              "AvailabilityZone": "us-west-1b"
          },
          {
              "Timestamp": "2014-01-06T07:10:55.000Z",
              "ProductDescription": "SUSE Linux",
              "InstanceType": "m1.xlarge",
              "SpotPrice": "0.087000",
              "AvailabilityZone": "us-west-1c"
          },
          {
              "Timestamp": "2014-01-06T05:42:36.000Z",
              "ProductDescription": "SUSE Linux (Amazon VPC)",
              "InstanceType": "m1.xlarge",
              "SpotPrice": "0.087000",
              "AvailabilityZone": "us-west-1a"
      },
      ...
}
```

Pour décrire l'historique des prix au comptant pour Linux/UNIX Amazon VPC

Cet exemple de commande renvoie l'historique des prix au comptant pour les instances Amazon Linux/UNIX VPC m1.xlarge pour un jour donné de janvier.

Commande :

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-
description "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-
time 2014-01-06T08:09:10
```

Sortie :

```
{
  "SpotPriceHistory": [
      {
          "Timestamp": "2014-01-06T04:32:53.000Z",
          "ProductDescription": "Linux/UNIX (Amazon VPC)",
          "InstanceType": "m1.xlarge",
          "SpotPrice": "0.080000",
          "AvailabilityZone": "us-west-1a"
      },
      {
          "Timestamp": "2014-01-05T11:28:26.000Z",
          "ProductDescription": "Linux/UNIX (Amazon VPC)",
          "InstanceType": "m1.xlarge",
          "SpotPrice": "0.080000",
          "AvailabilityZone": "us-west-1c"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSpotPriceHistoryà la section Référence des AWS CLI commandes.

`describe-stale-security-groups`

L'exemple de code suivant montre comment utiliser`describe-stale-security-groups`.

AWS CLI

Pour décrire les groupes de sécurité obsolètes

Cet exemple décrit les règles de groupe de sécurité obsolètes pour`vpc-11223344`. La réponse indique que le code sg-5fa68d3a de votre compte possède une règle SSH d'entrée périmée qui fait référence `sg-279ab042` au VPC homologue, et que votre compte possède une règle SSH de sortie obsolète `sg-fe6fba9a` qui fait référence au VPC homologue. `sg-ef6fba8b`

Commande :

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

Sortie :

```
{
  "StaleSecurityGroupSet": [
      {
          "VpcId": "vpc-11223344",
          "StaleIpPermissionsEgress": [
              {
                  "ToPort": 22,
                  "FromPort": 22,
                  "UserIdGroupPairs": [
                      {
                          "VpcId": "vpc-7a20e51f",
                          "GroupId": "sg-ef6fba8b",
                          "VpcPeeringConnectionId": "pcx-b04deed9",
                          "PeeringStatus": "active"
                      }
                  ],
                  "IpProtocol": "tcp"
              }
          ],
          "GroupName": "MySG1",
          "StaleIpPermissions": [],
          "GroupId": "sg-fe6fba9a",
          "Description": MySG1"
      },
      {
          "VpcId": "vpc-11223344",
          "StaleIpPermissionsEgress": [],
          "GroupName": "MySG2",
          "StaleIpPermissions": [
              {
                  "ToPort": 22,
                  "FromPort": 22,
                  "UserIdGroupPairs": [
                      {
                          "VpcId": "vpc-7a20e51f",
                          "GroupId": "sg-279ab042",
                          "Description": "Access from pcx-b04deed9",
                          "VpcPeeringConnectionId": "pcx-b04deed9",
                          "PeeringStatus": "active"
                      }
                  ],
                  "IpProtocol": "tcp"
              }
```

```
            ],
            "GroupId": "sg-5fa68d3a",
            "Description": "MySG2"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStaleSecurityGroupsà la section Référence des AWS CLI commandes.

`describe-store-image-tasks`

L'exemple de code suivant montre comment utiliser`describe-store-image-tasks`.

AWS CLI

Pour décrire la progression d'une tâche de stockage AMI

L'`describe-store-image-tasks`exemple suivant décrit la progression d'une tâche de stockage AMI.

```
aws ec2 describe-store-image-tasks
```

Sortie :

```
{
    "StoreImageTaskResults": [
        {
            "AmiId": "ami-1234567890abcdef0",
            "Bucket": "my-ami-bucket",
            "ProgressPercentage": 17,
            "S3objectKey": "ami-1234567890abcdef0.bin",
            "StoreTaskState": "InProgress",
            "StoreTaskFailureReason": null,
            "TaskStartTime": "2022-01-01T01:01:01.001Z"
        }
    ]
}
```

Pour plus d'informations sur le stockage et la restauration d'une AMI à l'aide de S3, consultez Stocker et restaurer une AMI à l'aide de S3 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html> dans le guide de l'utilisateur Amazon. EC2

- Pour plus de détails sur l'API, reportez-vous DescribeStoreImageTasksà la section Référence des AWS CLI commandes.

`describe-subnets`

L'exemple de code suivant montre comment utiliser`describe-subnets`.

AWS CLI

Exemple 1 : description de l'ensemble de vos sous-réseaux

L'exemple `describe-subnets` suivant affiche les détails de vos sous-réseaux.

```
aws ec2 describe-subnets
```

Sortie :

```
{
    "Subnets": [
        {
            "AvailabilityZone": "us-east-1d",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4089,
            "CidrBlock": "172.31.80.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": false,
            "MapCustomerOwnedIpOnLaunch": true,
            "State": "available",
            "SubnetId": "subnet-0bb1c79de3EXAMPLE",
            "VpcId": "vpc-0ee975135dEXAMPLE",
            "OwnerId": "111122223333",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "CustomerOwnedIpv4Pool:": 'pool-2EXAMPLE',
            "SubnetArn": "arn:aws:ec2:us-east-2:111122223333:subnet/
subnet-0bb1c79de3EXAMPLE",
            "EnableDns64": false,
            "Ipv6Native": false,
```

```
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        },
        {
            "AvailabilityZone": "us-east-1d",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4089,
            "CidrBlock": "172.31.80.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true,
            "MapCustomerOwnedIpOnLaunch": false,
            "State": "available",
            "SubnetId": "subnet-8EXAMPLE",
            "VpcId": "vpc-3EXAMPLE",
            "OwnerId": "1111222233333",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "MySubnet"
                }
            ],
            "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des sous-réseaux VPCs et des sous-réseaux dans le guide de l'AWS utilisateur VPC.

Exemple 2 : description des sous-réseaux d’un VPC spécifique

L'exemple `describe-subnets` suivant utilise un filtre pour récupérer les détails des sous-réseaux du VPC spécifié.

```
aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=vpc-3EXAMPLE"
```

Sortie :

```
{
    "Subnets": [
        {
            "AvailabilityZone": "us-east-1d",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4089,
            "CidrBlock": "172.31.80.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true,
            "MapCustomerOwnedIpOnLaunch": false,
            "State": "available",
            "SubnetId": "subnet-8EXAMPLE",
            "VpcId": "vpc-3EXAMPLE",
            "OwnerId": "1111222233333",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "MySubnet"
                }
            ],
            "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des sous-réseaux VPCs et des sous-réseaux dans le guide de l'AWS utilisateur VPC.

Exemple 3 : description des sous-réseaux avec une balise spécifique

L'`describe-subnets`exemple suivant utilise un filtre pour récupérer les détails des sous-réseaux avec la balise `CostCenter=123` et le `--query` paramètre pour afficher le sous-réseau IDs des sous-réseaux avec cette balise.

```
aws ec2 describe-subnets \
    --filters "Name=tag:CostCenter,Values=123" \
    --query "Subnets[*].SubnetId" \
    --output text
```

Sortie :

```
subnet-0987a87c8b37348ef
subnet-02a95061c45f372ee
subnet-03f720e7de2788d73
```

Pour plus d'informations, consultez la section Utilisation des sous-réseaux VPCs et des sous-réseaux dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DescribeSubnetsà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Exemple 1 : pour décrire toutes les balises d'une seule ressource

L'`describe-tags`exemple suivant décrit les balises de l'instance spécifiée.

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=i-1234567890abcdef8"
```

Sortie :

```
{
```

```
    "Tags": [
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Beta Server",
            "Key": "Name"
        }
    ]
}
```

Exemple 2 : pour décrire toutes les balises d'un type de ressource

L'`describe-tags`exemple suivant décrit les balises de vos volumes.

```
aws ec2 describe-tags \
    --filters "Name=resource-type,Values=volume"
```

Sortie :

```
{
    "Tags": [
        {
            "ResourceType": "volume",
            "ResourceId": "vol-1234567890abcdef0",
            "Value": "Project1",
            "Key": "Purpose"
        },
        {
            "ResourceType": "volume",
            "ResourceId": "vol-049df61146c4d7901",
            "Value": "Logs",
            "Key": "Purpose"
        }
    ]
}
```

Exemple 3 : Pour décrire tous vos tags

L'`describe-tags`exemple suivant décrit les balises de toutes vos ressources.

```
aws ec2 describe-tags
```

Exemple 4 : Pour décrire les balises de vos ressources en fonction d'une clé de balise

L'`describe-tags`exemple suivant décrit les balises de vos ressources dotées d'une balise avec la clé`Stack`.

```
aws ec2 describe-tags \
    --filters Name=key,Values=Stack
```

Sortie :

```
{
    "Tags": [
        {
            "ResourceType": "volume",
            "ResourceId": "vol-027552a73f021f3b",
            "Value": "Production",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        }
    ]
}
```

Exemple 5 : pour décrire les balises de vos ressources en fonction d'une clé de balise et d'une valeur de balise

L'`describe-tags`exemple suivant décrit les balises associées à vos ressources`Stack=Test`.

```
aws ec2 describe-tags \
    --filters Name=key,Values=Stack Name=value,Values=Test
```

Sortie :

```
{
```

```
    "Tags": [
        {
            "ResourceType": "image",
            "ResourceId": "ami-3ac336533f021f3bd",
            "Value": "Test",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        }
    ]
}
```

L'`describe-tags`exemple suivant utilise une syntaxe alternative pour décrire les ressources associées à la balise`Stack=Test`.

```
aws ec2 describe-tags \
    --filters "Name=tag:Stack,Values=Test"
```

L'`describe-tags`exemple suivant décrit les balises de toutes vos instances dont une balise comporte une clé `Purpose` mais aucune valeur.

```
aws ec2 describe-tags \
    --filters "Name=resource-
type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="
```

Sortie :

```
{
    "Tags": [
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef5",
            "Value": null,
            "Key": "Purpose"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`describe-traffic-mirror-filters`

L'exemple de code suivant montre comment utiliser`describe-traffic-mirror-filters`.

AWS CLI

Pour afficher les filtres de votre rétroviseur

L'`describe-traffic-mirror-filters`exemple suivant affiche les détails de tous vos filtres de rétroviseur du trafic.

```
aws ec2 describe-traffic-mirror-filters
```

Sortie :

```
{
    "TrafficMirrorFilters": [
        {
            "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
            "IngressFilterRules": [
                {
                    "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
                    "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
                    "TrafficDirection": "ingress",
                    "RuleNumber": 100,
                    "RuleAction": "accept",
                    "Protocol": 6,
                    "DestinationCidrBlock": "10.0.0.0/24",
                    "SourceCidrBlock": "10.0.0.0/24",
                    "Description": "TCP Rule"
                }
            ],
            "EgressFilterRules": [],
            "NetworkServices": [],
            "Description": "Example filter",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, voir Afficher vos filtres de miroir du trafic dans le guide Traffic Mirroring.

- Pour plus de détails sur l'API, reportez-vous DescribeTrafficMirrorFiltersà la section Référence des AWS CLI commandes.

`describe-traffic-mirror-sessions`

L'exemple de code suivant montre comment utiliser`describe-traffic-mirror-sessions`.

AWS CLI

Pour décrire une session Traffic Mirror

L'`describe-traffic-mirror-sessions`exemple suivant affiche les détails de vos sessions Traffic Mirror.

```
aws ec2 describe-traffic-mirror-sessions
```

Sortie :

```
{
    "TrafficMirrorSessions": [
        {
            "Tags": [],
            "VirtualNetworkId": 42,
            "OwnerId": "111122223333",
            "Description": "TCP Session",
            "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
            "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
            "TrafficMirrorFilterId": "tmf-083e18f985EXAMPLE",
            "PacketLength": 20,
            "SessionNumber": 1,
            "TrafficMirrorSessionId": "tms-0567a4c684EXAMPLE"
        },
        {
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "tag test"
                }
            ],
            "VirtualNetworkId": 13314501,
            "OwnerId": "111122223333",
```

```
            "Description": "TCP Session",
            "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
            "TrafficMirrorTargetId": "tmt-03665551cbEXAMPLE",
            "TrafficMirrorFilterId": "tmf-06c787846cEXAMPLE",
            "SessionNumber": 2,
            "TrafficMirrorSessionId": "tms-0060101cf8EXAMPLE"
        }
    ]
}
```

Pour plus d'informations, voir Afficher les détails de la session Traffic Mirror dans le guide AWS Traffic Mirroring.

- Pour plus de détails sur l'API, reportez-vous DescribeTrafficMirrorSessionsà la section Référence des AWS CLI commandes.

`describe-traffic-mirror-targets`

L'exemple de code suivant montre comment utiliser`describe-traffic-mirror-targets`.

AWS CLI

Pour décrire une cible miroir du trafic

L'`describe-traffic-mirror-targets`exemple suivant affiche des informations sur la cible de miroir de trafic spécifiée.

```
aws ec2 describe-traffic-mirror-targets \
    --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE
```

Sortie :

```
{
    "TrafficMirrorTargets": [
        {
            "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
            "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/net/NLB/7cdec873fEXAMPLE",
            "Type": "network-load-balancer",
            "Description": "Example Network Load Balancer target",
            "OwnerId": "111122223333",
            "Tags": []
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Traffic Mirror targets dans le guide Amazon VPC Traffic Mirroring.

- Pour plus de détails sur l'API, reportez-vous DescribeTrafficMirrorTargetsà la section Référence des AWS CLI commandes.

`describe-transit-gateway-attachments`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-attachments`.

AWS CLI

Pour consulter les pièces jointes de votre passerelle de transit

L'`describe-transit-gateway-attachments`exemple suivant affiche les détails des pièces jointes à votre passerelle de transit.

```
aws ec2 describe-transit-gateway-attachments
```

Sortie :

```
{
    "TransitGatewayAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "TransitGatewayOwnerId": "123456789012",
            "ResourceOwnerId": "123456789012",
            "ResourceType": "vpc",
            "ResourceId": "vpc-3EXAMPLE",
            "State": "available",
            "Association": {
                "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
                "State": "associated"
            },
            "CreationTime": "2019-08-26T14:59:25.000Z",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Example"
```

```
            }
        ]
    },
    {
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "TransitGatewayOwnerId": "123456789012",
        "ResourceOwnerId": "123456789012",
        "ResourceType": "vpc",
        "ResourceId": "vpc-0065acced4EXAMPLE",
        "State": "available",
        "Association": {
            "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
            "State": "associated"
        },
        "CreationTime": "2019-08-07T17:03:07.000Z",
        "Tags": []
    },
    {
        "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "TransitGatewayOwnerId": "123456789012",
        "ResourceOwnerId": "123456789012",
        "ResourceType": "direct-connect-gateway",
        "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "State": "available",
        "Association": {
            "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
            "State": "associated"
        },
        "CreationTime": "2019-08-14T20:27:44.000Z",
        "Tags": []
    },
    {
        "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "TransitGatewayOwnerId": "123456789012",
        "ResourceOwnerId": "123456789012",
        "ResourceType": "direct-connect-gateway",
        "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
        "State": "available",
        "Association": {
            "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
            "State": "associated"
```

```
            },
            "CreationTime": "2019-08-14T20:33:02.000Z",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, voir Travailler avec les passerelles de transport en commun dans le Guide des passerelles de transport en commun.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayAttachmentsà la section Référence des AWS CLI commandes.

`describe-transit-gateway-connect-peers`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-connect-peers`.

AWS CLI

Pour décrire un homologue Transit Gateway Connect

L'`describe-transit-gateway-connect-peers`exemple suivant décrit l'homologue Connect spécifié.

```
aws ec2 describe-transit-gateway-connect-peers \
    --transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE
```

Sortie :

```
{
    "TransitGatewayConnectPeers": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
            "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
            "State": "available",
            "CreationTime": "2021-10-13T03:35:17.000Z",
            "ConnectPeerConfiguration": {
                "TransitGatewayAddress": "10.0.0.234",
                "PeerAddress": "172.31.1.11",
                "InsideCidrBlocks": [
```

```
                    "169.254.6.0/29"
                ],
                "Protocol": "gre",
                "BgpConfigurations": [
                    {
                        "TransitGatewayAsn": 64512,
                        "PeerAsn": 64512,
                        "TransitGatewayAddress": "169.254.6.2",
                        "PeerAddress": "169.254.6.1",
                        "BgpStatus": "down"
                    },
                    {
                        "TransitGatewayAsn": 64512,
                        "PeerAsn": 64512,
                        "TransitGatewayAddress": "169.254.6.3",
                        "PeerAddress": "169.254.6.1",
                        "BgpStatus": "down"
                    }
                ]
            },
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez les pièces jointes Transit Gateway Connect et les homologues Transit Gateway Connect dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayConnectPeersà la section Référence des AWS CLI commandes.

`describe-transit-gateway-connects`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-connects`.

AWS CLI

Pour décrire une passerelle de transit, connectez une pièce jointe

L'`describe-transit-gateway-connects`exemple suivant décrit la pièce jointe Connect spécifiée.

```
aws ec2 describe-transit-gateway-connects \
```

```
    --transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

Sortie :

```
{
    "TransitGatewayConnects": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
            "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "State": "available",
            "CreationTime": "2021-03-09T19:59:17+00:00",
            "Options": {
                "Protocol": "gre"
            },
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez les pièces jointes Transit Gateway Connect et les homologues Transit Gateway Connect dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayConnectsà la section Référence des AWS CLI commandes.

`describe-transit-gateway-multicast-domains`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-multicast-domains`.

AWS CLI

Pour décrire les domaines de multidiffusion de votre passerelle de transit

L'`describe-transit-gateway-multicast-domains`exemple suivant affiche les détails de tous les domaines de multidiffusion de votre passerelle de transit.

```
aws ec2 describe-transit-gateway-multicast-domains
```

Sortie :

```
{

    "TransitGatewayMulticastDomains": [
        {
            "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
            "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
            "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
east-1:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
            "OwnerId": "123456789012",
            "Options": {
                "Igmpv2Support": "disable",
                "StaticSourcesSupport": "enable",
                "AutoAcceptSharedAssociations": "disable"
            },
            "State": "available",
            "CreationTime": "2019-12-10T18:32:50+00:00",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "mc1"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des domaines de multidiffusion dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayMulticastDomainsà la section Référence des AWS CLI commandes.

`describe-transit-gateway-peering-attachments`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-peering-attachments`.

AWS CLI

Pour décrire les pièces jointes de peering de votre passerelle de transit

L'`describe-transit-gateway-peering-attachments`exemple suivant affiche les détails de toutes les pièces jointes de peering de votre passerelle de transit.

```
aws ec2 describe-transit-gateway-peering-attachments
```

Sortie :

```
{
    "TransitGatewayPeeringAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
            "RequesterTgwInfo": {
                "TransitGatewayId": "tgw-123abc05e04123abc",
                "OwnerId": "123456789012",
                "Region": "us-west-2"
            },
            "AccepterTgwInfo": {
                "TransitGatewayId": "tgw-11223344aabbcc112",
                "OwnerId": "123456789012",
                "Region": "us-east-2"
            },
            "State": "pendingAcceptance",
            "CreationTime": "2019-12-09T11:38:05.000Z",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Transit Gateway peering Attachments dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayPeeringAttachmentsà la section Référence des AWS CLI commandes.

`describe-transit-gateway-policy-tables`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-policy-tables`.

AWS CLI

Pour décrire un tableau des politiques relatives aux passerelles de transit

L'`describe-transit-gateway-policy-tables`exemple suivant décrit la table de politique de passerelle de transit spécifiée.

```
aws ec2 describe-transit-gateway-policy-tables \
    --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

Sortie :

```
{
    "TransitGatewayPolicyTables": [
        {
            "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
            "TransitGatewayId": "tgw-067f8505c18f0bd6e",
            "State": "available",
            "CreationTime": "2023-11-28T16:36:43+00:00",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez les tableaux des politiques relatives aux passerelles de transit dans le guide de l'utilisateur de Transit Gateway.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayPolicyTablesà la section Référence des AWS CLI commandes.

`describe-transit-gateway-route-tables`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-route-tables`.

AWS CLI

Pour décrire les tables de routage de votre passerelle de transit

L'`describe-transit-gateway-route-tables`exemple suivant affiche les détails des tables de routage de votre passerelle de transit.

```
aws ec2 describe-transit-gateway-route-tables
```

Sortie :

```
{
    "TransitGatewayRouteTables": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0ca78a549EXAMPLE",
            "TransitGatewayId": "tgw-0bc994abffEXAMPLE",
            "State": "available",
            "DefaultAssociationRouteTable": true,
            "DefaultPropagationRouteTable": true,
            "CreationTime": "2018-11-28T14:24:49.000Z",
            "Tags": []
        },
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0e8f48f148EXAMPLE",
            "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
            "State": "available",
            "DefaultAssociationRouteTable": true,
            "DefaultPropagationRouteTable": true,
            "CreationTime": "2018-11-28T14:24:00.000Z",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, voir Afficher les tables de routage des passerelles de transit dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayRouteTablesà la section Référence des AWS CLI commandes.

`describe-transit-gateway-vpc-attachments`

L'exemple de code suivant montre comment utiliser`describe-transit-gateway-vpc-attachments`.

AWS CLI

Pour décrire les pièces jointes VPC de votre passerelle de transit

L'`describe-transit-gateway-vpc-attachments`exemple suivant affiche les détails des pièces jointes VPC de votre passerelle de transit.

```
aws ec2 describe-transit-gateway-vpc-attachments
```

Sortie :

```
{
    "TransitGatewayVpcAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-0a08e88308EXAMPLE",
            "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
            "VpcId": "vpc-0f501f7ee8EXAMPLE",
            "VpcOwnerId": "111122223333",
            "State": "available",
            "SubnetIds": [
                "subnet-045d586432EXAMPLE",
                "subnet-0a0ad478a6EXAMPLE"
            ],
            "CreationTime": "2019-02-13T11:04:02.000Z",
            "Options": {
                "DnsSupport": "enable",
                "Ipv6Support": "disable"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "attachment name"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher vos pièces jointes VPC dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewayVpcAttachmentsà la section Référence des AWS CLI commandes.

`describe-transit-gateways`

L'exemple de code suivant montre comment utiliser`describe-transit-gateways`.

AWS CLI

Pour décrire vos passerelles de transport en commun

L'`describe-transit-gateways`exemple suivant permet de récupérer des informations sur vos passerelles de transport en commun.

```
aws ec2 describe-transit-gateways
```

Sortie :

```
{
    "TransitGateways": [
        {
            "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
            "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0262a0e521EXAMPLE",
            "State": "available",
            "OwnerId": "111122223333",
            "Description": "MyTGW",
            "CreationTime": "2019-07-10T14:02:12.000Z",
            "Options": {
                "AmazonSideAsn": 64516,
                "AutoAcceptSharedAttachments": "enable",
                "DefaultRouteTableAssociation": "enable",
                "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
                "DefaultRouteTablePropagation": "enable",
                "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
                "VpnEcmpSupport": "enable",
                "DnsSupport": "enable"
            },
            "Tags": []
        },
        {
            "TransitGatewayId": "tgw-0fb8421e2dEXAMPLE",
            "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0fb8421e2da853bf3",
            "State": "available",
            "OwnerId": "111122223333",
            "CreationTime": "2019-03-15T22:57:33.000Z",
            "Options": {
                "AmazonSideAsn": 65412,
                "AutoAcceptSharedAttachments": "disable",
                "DefaultRouteTableAssociation": "enable",
                "AssociationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
                "DefaultRouteTablePropagation": "enable",
                "PropagationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
```

```
                "VpnEcmpSupport": "enable",
                "DnsSupport": "enable"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "TGW1"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTransitGatewaysà la section Référence des AWS CLI commandes.

`describe-verified-access-endpoints`

L'exemple de code suivant montre comment utiliser`describe-verified-access-endpoints`.

AWS CLI

Pour décrire un point de terminaison d'accès vérifié

L'`describe-verified-access-endpoints`exemple suivant décrit le point de terminaison Verified Access spécifié.

```
aws ec2 describe-verified-access-endpoints \
    --verified-access-endpoint-ids vae-066fac616d4d546f2
```

Sortie :

```
{
    "VerifiedAccessEndpoints": [
        {
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
            "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
            "ApplicationDomain": "example.com",
            "EndpointType": "network-interface",
            "AttachmentType": "vpc",
```

```
            "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
            "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
            "SecurityGroupIds": [
                "sg-004915970c4c8f13a"
            ],
            "NetworkInterfaceOptions": {
                "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
                "Protocol": "https",
                "Port": 443
            },
            "Status": {
                "Code": "active"
            },
            "Description": "",
            "CreationTime": "2023-08-25T20:54:43",
            "LastUpdatedTime": "2023-08-25T22:17:26",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-va-endpoint"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Points de terminaison Verified Access dans le Guide de l'utilisateur AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DescribeVerifiedAccessEndpointsà la section Référence des AWS CLI commandes.

`describe-verified-access-groups`

L'exemple de code suivant montre comment utiliser`describe-verified-access-groups`.

AWS CLI

Pour décrire un groupe d'accès vérifié

L'`describe-verified-access-groups`exemple suivant décrit le groupe d'accès vérifié spécifié.

```
aws ec2 describe-verified-access-groups \
    --verified-access-group-ids vagr-0dbe967baf14b7235
```

Sortie :

```
{
    "VerifiedAccessGroups": [
        {
            "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "Description": "Testing Verified Access",
            "Owner": "123456789012",
            "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
            "CreationTime": "2023-08-25T19:55:19",
            "LastUpdatedTime": "2023-08-25T22:17:25",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-va-group"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DescribeVerifiedAccessGroupsà la section Référence des AWS CLI commandes.

`describe-verified-access-instance-logging-configurations`

L'exemple de code suivant montre comment utiliser`describe-verified-access-instance-logging-configurations`.

AWS CLI

Pour décrire la configuration de journalisation pour une instance d'accès vérifié

L'`describe-verified-access-instance-logging-configurations`exemple suivant décrit la configuration de journalisation pour l'instance Verified Access spécifiée.

```
aws ec2 describe-verified-access-instance-logging-configurations \
    --verified-access-instance-ids vai-0ce000c0b7643abea
```

Sortie :

```
{
    "LoggingConfigurations": [
        {
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "AccessLogs": {
                "S3": {
                    "Enabled": false
                },
                "CloudWatchLogs": {
                    "Enabled": true,
                    "DeliveryStatus": {
                        "Code": "success"
                    },
                    "LogGroup": "my-log-group"
                },
                "KinesisDataFirehose": {
                    "Enabled": false
                },
                "LogVersion": "ocsf-1.0.0-rc.2",
                "IncludeTrustContext": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Journaux d'accès vérifiés dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DescribeVerifiedAccessInstanceLoggingConfigurationsà la section Référence des AWS CLI commandes.

`describe-verified-access-instances`

L'exemple de code suivant montre comment utiliser`describe-verified-access-instances`.

AWS CLI

Pour décrire une instance d'accès vérifié

L'`describe-verified-access-instances`exemple suivant décrit l'instance Verified Access spécifiée.

```
aws ec2 describe-verified-access-instances \
    --verified-access-instance-ids vai-0ce000c0b7643abea
```

Sortie :

```
{
    "VerifiedAccessInstances": [
        {
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "Description": "Testing Verified Access",
            "VerifiedAccessTrustProviders": [
                {
                    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
                    "TrustProviderType": "user",
                    "UserTrustProviderType": "iam-identity-center"
                }
            ],
            "CreationTime": "2023-08-25T18:27:56",
            "LastUpdatedTime": "2023-08-25T19:03:32",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-ava-instance"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Instances d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DescribeVerifiedAccessInstancesà la section Référence des AWS CLI commandes.

`describe-verified-access-trust-providers`

L'exemple de code suivant montre comment utiliser`describe-verified-access-trust-providers`.

AWS CLI

Pour décrire un fournisseur de confiance Verified Access

L'`describe-verified-access-trust-providers`exemple suivant décrit le fournisseur de confiance Verified Access spécifié.

```
aws ec2 describe-verified-access-trust-providers \
    --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

Sortie :

```
{
    "VerifiedAccessTrustProviders": [
        {
            "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
            "Description": "Testing Verified Access",
            "TrustProviderType": "user",
            "UserTrustProviderType": "iam-identity-center",
            "PolicyReferenceName": "idc",
            "CreationTime": "2023-08-25T19:00:38",
            "LastUpdatedTime": "2023-08-25T19:03:32",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-va-trust-provider"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Trust providers for Verified Access dans le Guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DescribeVerifiedAccessTrustProvidersà la section Référence des AWS CLI commandes.

`describe-volume-attribute`

L'exemple de code suivant montre comment utiliser`describe-volume-attribute`.

AWS CLI

Pour décrire un attribut de volume

Cet exemple de commande décrit l'`autoEnableIo`attribut du volume avec l'ID`vol-049df61146c4d7901`.

Commande :

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --
attribute autoEnableIO
```

Sortie :

```
{
    "AutoEnableIO": {
        "Value": false
    },
    "VolumeId": "vol-049df61146c4d7901"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVolumeAttributeà la section Référence des AWS CLI commandes.

`describe-volume-status`

L'exemple de code suivant montre comment utiliser`describe-volume-status`.

AWS CLI

Pour décrire l'état d'un volume unique

Cet exemple de commande décrit l'état du volume`vol-1234567890abcdef0`.

Commande :

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

Sortie :

```
{
    "VolumeStatuses": [
        {
            "VolumeStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "io-enabled"
                    },
                    {
                        "Status": "not-applicable",
                        "Name": "io-performance"
                    }
                ]
            },
            "AvailabilityZone": "us-east-1a",
            "VolumeId": "vol-1234567890abcdef0",
            "Actions": [],
            "Events": []
        }
    ]
}
```

Pour décrire l'état des volumes altérés

Cet exemple de commande décrit l'état de tous les volumes endommagés. Dans cet exemple de sortie, aucun volume n'est altéré.

Commande :

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

Sortie :

```
{
```

```
    "VolumeStatuses": []
}
```

Si la vérification de l'état d'un volume a échoué (l'état est altéré), consultez la section Travailler avec un volume altéré dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeVolumeStatusà la section Référence des AWS CLI commandes.

`describe-volumes-modifications`

L'exemple de code suivant montre comment utiliser`describe-volumes-modifications`.

AWS CLI

Pour décrire l'état de modification d'un volume

L'`describe-volumes-modifications`exemple suivant décrit l'état de modification du volume spécifié.

```
aws ec2 describe-volumes-modifications \
    --volume-ids vol-1234567890abcdef0
```

Sortie :

```
{
    "VolumeModification": {
        "TargetSize": 150,
        "TargetVolumeType": "io1",
        "ModificationState": "optimizing",
        "VolumeId": " vol-1234567890abcdef0",
        "TargetIops": 100,
        "StartTime": "2019-05-17T11:27:19.000Z",
        "Progress": 70,
        "OriginalVolumeType": "io1",
        "OriginalIops": 100,
        "OriginalSize": 100
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVolumesModificationsà la section Référence des AWS CLI commandes.

`describe-volumes`

L'exemple de code suivant montre comment utiliser`describe-volumes`.

AWS CLI

Exemple 1 : pour décrire un volume

L'`describe-volumes`exemple suivant décrit les volumes spécifiés dans la région actuelle.

```
aws ec2 describe-volumes \
    --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

Sortie :

```
{
    "Volumes": [
        {
            "AvailabilityZone": "us-east-1a",
            "Attachments": [
                {
                    "AttachTime": "2013-12-18T22:35:00.000Z",
                    "InstanceId": "i-1234567890abcdef0",
                    "VolumeId": "vol-049df61146c4d7901",
                    "State": "attached",
                    "DeleteOnTermination": true,
                    "Device": "/dev/sda1"
                }
            ],
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-2a:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513eEXAMPLE,
            "VolumeType": "gp2",
            "VolumeId": "vol-049df61146c4d7901",
            "State": "in-use",
            "Iops": 100,
            "SnapshotId": "snap-1234567890abcdef0",
            "CreateTime": "2019-12-18T22:35:00.084Z",
            "Size": 8
        },
        {
            "AvailabilityZone": "us-east-1a",
            "Attachments": [],
            "Encrypted": false,
```

```
            "VolumeType": "gp2",
            "VolumeId": "vol-1234567890abcdef0",
            "State": "available",
            "Iops": 300,
            "SnapshotId": "",
            "CreateTime": "2020-02-27T00:02:41.791Z",
            "Size": 100
        }
    ]
}
```

Exemple 2 : pour décrire les volumes attachés à une instance spécifique

L'`describe-volumes`exemple suivant décrit tous les volumes qui sont à la fois attachés à l'instance spécifiée et définis pour être supprimés lorsque l'instance se termine.

```
aws ec2 describe-volumes \
    --region us-east-1 \
    --filters Name=attachment.instance-
id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true
```

Pour obtenir un exemple de la manière de procéder pour `describe-volumes`, reportez-vous à l'exemple 1.

Exemple 3 : pour décrire les volumes disponibles dans une zone de disponibilité spécifique

L'`describe-volumes`exemple suivant décrit tous les volumes dont le statut est `available` et se trouvent dans la zone de disponibilité spécifiée.

```
aws ec2 describe-volumes \
    --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a
```

Pour obtenir un exemple de la manière de procéder pour `describe-volumes`, reportez-vous à l'exemple 1.

Exemple 4 : pour décrire des volumes en fonction de balises

L'`describe-volumes`exemple suivant décrit tous les volumes dotés de la clé de balise `Name` et d'une valeur commençant par`Test`. La sortie est ensuite filtrée avec une requête qui affiche uniquement les balises et IDs les volumes.

```
aws ec2 describe-volumes \
```

```
    --filters Name=tag:Name,Values=Test* \
    --query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

Sortie :

```
[
    {
        "Tag": [
            {
                "Value": "Test2",
                "Key": "Name"
            }
        ],
        "ID": "vol-1234567890abcdef0"
    },
    {
        "Tag": [
            {
                "Value": "Test1",
                "Key": "Name"
            }
        ],
        "ID": "vol-049df61146c4d7901"
    }
]
```

Pour d'autres exemples d'utilisation de filtres de balises, consultez la section Utilisation des balises dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeVolumesà la section Référence des AWS CLI commandes.

`describe-vpc-attribute`

L'exemple de code suivant montre comment utiliser`describe-vpc-attribute`.

AWS CLI

Pour décrire l' enableDnsSupport attribut

Cet exemple décrit l'`enableDnsSupport`attribut. Cet attribut indique si la résolution DNS est activée pour le VPC. Si cet attribut est `true`, le serveur Amazon DNS résout les noms d'hôte DNS

de vos instances par rapport aux adresses IP correspondantes. Dans le cas contraire, il ne le fait pas.

Commande :

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

Sortie :

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsSupport": {
        "Value": true
    }
}
```

Pour décrire l' enableDnsHostnames attribut

Cet exemple décrit l'`enableDnsHostnames`attribut. Cet attribut indique si les instances lancées dans le VPC obtiennent des noms d'hôte DNS. Si l'attribut est défini sur `true`, les instances incluses dans le VPC ont obtenu des noms d'hôte DNS.

Commande :

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

Sortie :

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsHostnames": {
        "Value": true
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcAttributeà la section Référence des AWS CLI commandes.

`describe-vpc-classic-link-dns-support`

L'exemple de code suivant montre comment utiliser`describe-vpc-classic-link-dns-support`.

AWS CLI

Pour décrire la prise en charge du ClassicLink DNS pour votre VPCs

Cet exemple décrit l'état de support ClassicLink DNS de tous vos VPCs.

Commande :

```
aws ec2 describe-vpc-classic-link-dns-support
```

Sortie :

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcClassicLinkDnsSupportà la section Référence des AWS CLI commandes.

`describe-vpc-classic-link`

L'exemple de code suivant montre comment utiliser`describe-vpc-classic-link`.

AWS CLI

Pour décrire l' ClassicLink état de votre VPCs

Cet exemple répertorie le ClassicLink statut de vpc-88888888.

Commande :

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

Sortie :

```
{
  "Vpcs": [
    {
      "ClassicLinkEnabled": true,
      "VpcId": "vpc-88888888",
      "Tags": [
        {
          "Value": "classiclinkvpc",
          "Key": "Name"
        }
      ]
    }
  ]
}
```

Cet exemple répertorie uniquement ceux VPCs qui sont activés pour Classiclink (la valeur du filtre `is-classic-link-enabled` est définie sur`true`).

Commande :

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-
enabled,Values=true"
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcClassicLinkà la section Référence des AWS CLI commandes.

`describe-vpc-endpoint-associations`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoint-associations`.

AWS CLI

Pour décrire les associations de points de terminaison VPC

L'`describe-vpc-endpoint-associations`exemple suivant décrit vos associations de points de terminaison VPC.

```
aws ec2 describe-vpc-endpoint-associations
```

Sortie :

```
{
    "VpcEndpointAssociations": [
        {
            "Id": "vpce-rsc-asc-0a810ca6ac8866bf9",
            "VpcEndpointId": "vpce-019b90d6f16d4f958",
            "AssociatedResourceAccessibility": "Accessible",
            "DnsEntry": {
                "DnsName":
 "vpce-019b90d6f16d4f958.rcfg-07129f3acded87625.4232ccc.vpc-lattice-rsc.us-
east-2.on.aws",
                "HostedZoneId": "Z03265862FOUNWMZOKUF4"
            },
            "AssociatedResourceArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les associations de points de terminaison VPC dans le guide de l'AWS PrivateLink utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointAssociationsà la section Référence des AWS CLI commandes.

`describe-vpc-endpoint-connection-notifications`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoint-connection-notifications`.

AWS CLI

Pour décrire les notifications de connexion des terminaux

L'`describe-vpc-endpoint-connection-notifications`exemple suivant décrit toutes les notifications de connexion de votre terminal.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

Sortie :

```
{
   "ConnectionNotificationSet": [
       {
           "ConnectionNotificationState": "Enabled",
           "ConnectionNotificationType": "Topic",
           "ConnectionEvents": [
               "Accept",
               "Reject",
               "Delete",
               "Connect"
           ],
           "ConnectionNotificationId": "vpce-nfn-04bcb952bc8af7abc",
           "ConnectionNotificationArn": "arn:aws:sns:us-
east-1:123456789012:VpceNotification",
           "VpcEndpointId": "vpce-0324151a02f327123"
       }
   ]
 }
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointConnectionNotificationsà la section Référence des AWS CLI commandes.

`describe-vpc-endpoint-connections`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoint-connections`.

AWS CLI

Pour décrire les connexions des points de terminaison VPC

Cet exemple décrit les connexions des points de terminaison d'interface à votre service de point de terminaison et filtre les résultats pour afficher les points de terminaison qui le sont`PendingAcceptance`.

Commande :

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

Sortie :

```
{
  "VpcEndpointConnections": [
      {
          "VpcEndpointId": "vpce-0abed31004e618123",
          "ServiceId": "vpce-svc-0abced088d20def56",
          "CreationTimestamp": "2017-11-30T10:00:24.350Z",
          "VpcEndpointState": "pendingAcceptance",
          "VpcEndpointOwner": "123456789012"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointConnectionsà la section Référence des AWS CLI commandes.

`describe-vpc-endpoint-service-configurations`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoint-service-configurations`.

AWS CLI

Pour décrire les configurations des services de point de terminaison

L'`describe-vpc-endpoint-service-configurations`exemple suivant décrit les configurations de vos services de point de terminaison.

```
aws ec2 describe-vpc-endpoint-service-configurations
```

Sortie :

```
{
    "ServiceConfigurations": [
        {
            "ServiceType": [
                {
                    "ServiceType": "GatewayLoadBalancer"
                }
            ],
            "ServiceId": "vpce-svc-012d33a1c4321cabc",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-012d33a1c4321cabc",
```

```
            "ServiceState": "Available",
            "AvailabilityZones": [
                "us-east-1d"
            ],
            "AcceptanceRequired": false,
            "ManagesVpcEndpoints": false,
            "GatewayLoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
gwy/GWLBService/123210844e429123"
            ],
            "Tags": []
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "ServiceId": "vpce-svc-123cabc125efa123",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123cabc125efa123",
            "ServiceState": "Available",
            "AvailabilityZones": [
                "us-east-1a"
            ],
            "AcceptanceRequired": true,
            "ManagesVpcEndpoints": false,
            "NetworkLoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
net/NLBforService/1238753950b25123"
            ],
            "BaseEndpointDnsNames": [
                "vpce-svc-123cabc125efa123.us-east-1.vpce.amazonaws.com"
            ],
            "PrivateDnsName": "example.com",
            "PrivateDnsNameConfiguration": {
                "State": "failed",
                "Type": "TXT",
                "Value": "vpce:qUAth3FdeABCApUiXabc",
                "Name": "_1d367jvbg34znqvyefrj"
            },
            "Tags": []
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Concepts du guide de AWS PrivateLink l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointServiceConfigurationsà la section Référence des AWS CLI commandes.

`describe-vpc-endpoint-service-permissions`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoint-service-permissions`.

AWS CLI

Pour décrire les autorisations des services de point de terminaison

Cet exemple décrit les autorisations pour le service de point de terminaison spécifié.

Commande :

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
```

Sortie :

```
{
   "AllowedPrincipals": [
       {
           "PrincipalType": "Account",
           "Principal": "arn:aws:iam::123456789012:root"
       }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointServicePermissionsà la section Référence des AWS CLI commandes.

`describe-vpc-endpoint-services`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoint-services`.

AWS CLI

Exemple 1 : pour décrire tous les services de point de terminaison VPC

L'`describe-vpc-endpoint-services`exemple suivant répertorie tous les services de point de terminaison VPC pour une AWS région.

```
aws ec2 describe-vpc-endpoint-services
```

Sortie :

```
{
    "ServiceDetails": [
        {
            "ServiceType": [
                {
                    "ServiceType": "Gateway"
                }
            ],
            "AcceptanceRequired": false,
            "ServiceName": "com.amazonaws.us-east-1.dynamodb",
            "VpcEndpointPolicySupported": true,
            "Owner": "amazon",
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
                "us-east-1f"
            ],
            "BaseEndpointDnsNames": [
                "dynamodb.us-east-1.amazonaws.com"
            ]
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "PrivateDnsName": "ec2.us-east-1.amazonaws.com",
            "ServiceName": "com.amazonaws.us-east-1.ec2",
```

```
            "VpcEndpointPolicySupported": false,
            "Owner": "amazon",
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
                "us-east-1f"
            ],
            "AcceptanceRequired": false,
            "BaseEndpointDnsNames": [
                "ec2.us-east-1.vpce.amazonaws.com"
            ]
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "PrivateDnsName": "ssm.us-east-1.amazonaws.com",
            "ServiceName": "com.amazonaws.us-east-1.ssm",
            "VpcEndpointPolicySupported": true,
            "Owner": "amazon",
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e"
            ],
            "AcceptanceRequired": false,
            "BaseEndpointDnsNames": [
                "ssm.us-east-1.vpce.amazonaws.com"
            ]
        }
    ],
    "ServiceNames": [
        "com.amazonaws.us-east-1.dynamodb",
        "com.amazonaws.us-east-1.ec2",
        "com.amazonaws.us-east-1.ec2messages",
        "com.amazonaws.us-east-1.elasticloadbalancing",
        "com.amazonaws.us-east-1.kinesis-streams",
```

```
        "com.amazonaws.us-east-1.s3",
        "com.amazonaws.us-east-1.ssm"
    ]
}
```

Exemple 2 : pour décrire les détails d'un service de point de terminaison

L'`describe-vpc-endpoint-services`exemple suivant répertorie les détails du service de point de terminaison de l'interface Amazon S3.

```
aws ec2 describe-vpc-endpoint-services \
    --filter 'Name=service-type,Values=Interface' Name=service-
name,Values=com.amazonaws.us-east-1.s3
```

Sortie :

```
{
    "ServiceDetails": [
        {
            "ServiceName": "com.amazonaws.us-east-1.s3",
            "ServiceId": "vpce-svc-081d84efcdEXAMPLE",
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
            "us-east-1f"
            ],
            "Owner": "amazon",
            "BaseEndpointDnsNames": [
                "s3.us-east-1.vpce.amazonaws.com"
            ],
            "VpcEndpointPolicySupported": true,
            "AcceptanceRequired": false,
            "ManagesVpcEndpoints": false,
            "Tags": []
        }
```

```
    ],
    "ServiceNames": [
        "com.amazonaws.us-east-1.s3"
    ]
}
```

Pour plus d'informations, voir Afficher les noms des AWS services disponibles dans le Guide de AWS PrivateLink l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointServicesà la section Référence des AWS CLI commandes.

`describe-vpc-endpoints`

L'exemple de code suivant montre comment utiliser`describe-vpc-endpoints`.

AWS CLI

Pour décrire vos points de terminaison VPC

L'`describe-vpc-endpoints`exemple suivant affiche les détails de tous vos points de terminaison VPC.

```
aws ec2 describe-vpc-endpoints
```

Sortie :

```
{
    "VpcEndpoints": [
        {
            "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}",
            "VpcId": "vpc-aabb1122",
            "NetworkInterfaceIds": [],
            "SubnetIds": [],
            "PrivateDnsEnabled": true,
            "State": "available",
            "ServiceName": "com.amazonaws.us-east-1.dynamodb",
            "RouteTableIds": [
                "rtb-3d560345"
            ],
            "Groups": [],
            "VpcEndpointId": "vpce-032a826a",
```

```
            "VpcEndpointType": "Gateway",
            "CreationTimestamp": "2017-09-05T20:41:28Z",
            "DnsEntries": [],
            "OwnerId": "123456789012"
        },
        {
            "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*
\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource
\": \"*\"\n    }\n  ]\n}",
            "VpcId": "vpc-1a2b3c4d",
            "NetworkInterfaceIds": [
                "eni-2ec2b084",
                "eni-1b4a65cf"
            ],
            "SubnetIds": [
                "subnet-d6fcaa8d",
                "subnet-7b16de0c"
            ],
            "PrivateDnsEnabled": false,
            "State": "available",
            "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
            "RouteTableIds": [],
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-54e8bf31"
                }
            ],
            "VpcEndpointId": "vpce-0f89a33420c1931d7",
            "VpcEndpointType": "Interface",
            "CreationTimestamp": "2017-09-05T17:55:27.583Z",
            "DnsEntries": [
                {
                    "HostedZoneId": "Z7HUB22UULQXV",
                    "DnsName": "vpce-0f89a33420c1931d7-
bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
                },
                {
                    "HostedZoneId": "Z7HUB22UULQXV",
                    "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1b.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
                },
                {
                    "HostedZoneId": "Z7HUB22UULQXV",
```

```
                    "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
                }
            ],
            "OwnerId": "123456789012"
        },
        {
            "VpcEndpointId": "vpce-aabbaabbaabbaabba",
            "VpcEndpointType": "GatewayLoadBalancer",
            "VpcId": "vpc-111122223333aabbc",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-123123a1c43abc123",
            "State": "available",
            "SubnetIds": [
                "subnet-0011aabbcc2233445"
            ],
            "RequesterManaged": false,
            "NetworkInterfaceIds": [
                "eni-01010120203030405"
            ],
            "CreationTimestamp": "2020-11-11T08:06:03.522Z",
            "Tags": [],
            "OwnerId": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez la section Concepts du guide de AWS PrivateLink l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeVpcEndpointsà la section Référence des AWS CLI commandes.

`describe-vpc-peering-connections`

L'exemple de code suivant montre comment utiliser`describe-vpc-peering-connections`.

AWS CLI

Pour décrire vos connexions d'appairage VPC

Cet exemple décrit toutes vos connexions d'appairage VPC.

Commande :

```
aws ec2 describe-vpc-peering-connections
```

Sortie :

```
{
    "VpcPeeringConnections": [
        {
            "Status": {
                "Message": "Active",
                "Code": "active"
            },
            "Tags": [
                {
                    "Value": "Peering-1",
                    "Key": "Name"
                }
            ],
            "AccepterVpcInfo": {
                "OwnerId": "111122223333",
                "VpcId": "vpc-1a2b3c4d",
                "CidrBlock": "10.0.1.0/28"
            },
            "VpcPeeringConnectionId": "pcx-11122233",
            "RequesterVpcInfo": {
                "PeeringOptions": {
                    "AllowEgressFromLocalVpcToRemoteClassicLink": false,
                    "AllowEgressFromLocalClassicLinkToRemoteVpc": false
                },
                "OwnerId": "444455556666",
                "VpcId": "vpc-123abc45",
                "CidrBlock": "192.168.0.0/16"
            }
        },
        {
            "Status": {
                "Message": "Pending Acceptance by 444455556666",
                "Code": "pending-acceptance"
            },
            "Tags": [],
            "RequesterVpcInfo": {
                "PeeringOptions": {
                    "AllowEgressFromLocalVpcToRemoteClassicLink": false,
                    "AllowEgressFromLocalClassicLinkToRemoteVpc": false
```

```
                },
                "OwnerId": "444455556666",
                "VpcId": "vpc-11aa22bb",
                "CidrBlock": "10.0.0.0/28"
            },
            "VpcPeeringConnectionId": "pcx-abababab",
            "ExpirationTime": "2014-04-03T09:12:43.000Z",
            "AccepterVpcInfo": {
                "OwnerId": "444455556666",
                "VpcId": "vpc-33cc44dd"
            }
        }
    ]
}
```

Pour décrire des connexions d'appairage VPC spécifiques

Cet exemple décrit toutes vos connexions d'appairage VPC dont l'état est en attente d'acceptation.

Commande :

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

Cet exemple décrit toutes vos connexions d'appairage VPC dotées du tag Owner=Finance.

Commande :

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

Cet exemple décrit toutes les connexions d'appairage VPC que vous avez demandées pour le VPC spécifié, vpc-1a2b3c4d.

Commande :

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcPeeringConnectionsà la section Référence des AWS CLI commandes.

describe-vpcs

L'exemple de code suivant montre comment utiliserdescribe-vpcs.

AWS CLI

Exemple 1 : Pour décrire tous vos VPCs

L'describe-vpcsexemple suivant permet de récupérer des informations sur votre VPCs.

```
aws ec2 describe-vpcs
```

Sortie :

```
{
    "Vpcs": [
        {
            "CidrBlock": "30.1.0.0/16",
            "DhcpOptionsId": "dopt-19edf471",
            "State": "available",
            "VpcId": "vpc-0e9801d129EXAMPLE",
            "OwnerId": "111122223333",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-062c64cfafEXAMPLE",
                    "CidrBlock": "30.1.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Not Shared"
                }
            ]
        },
        {
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-19edf471",
```

```
            "State": "available",
            "VpcId": "vpc-06e4ab6c6cEXAMPLE",
            "OwnerId": "222222222222",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Shared VPC"
                }
            ]
        }
    ]
}
```

Exemple 2 : description d'un VPC spécifié

L'exemple `describe-vpcs` suivant récupère les détails du VPC spécifié.

```
aws ec2 describe-vpcs \
    --vpc-ids vpc-06e4ab6c6cEXAMPLE
```

Sortie :

```
{
    "Vpcs": [
        {
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-19edf471",
            "State": "available",
            "VpcId": "vpc-06e4ab6c6cEXAMPLE",
            "OwnerId": "111122223333",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
```

```
                {
                    "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Shared VPC"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpcsà la section Référence des AWS CLI commandes.

`describe-vpn-connections`

L'exemple de code suivant montre comment utiliser`describe-vpn-connections`.

AWS CLI

Exemple 1 : Pour décrire vos connexions VPN

L'`describe-vpn-connections`exemple suivant décrit toutes vos connexions Site-to-Site VPN.

```
aws ec2 describe-vpn-connections
```

Sortie :

```
{
    "VpnConnections": [
        {
            "CustomerGatewayConfiguration": "...configuration information...",
            "CustomerGatewayId": "cgw-01234567abcde1234",
```

```
            "Category": "VPN",
            "State": "available",
            "Type": "ipsec.1",
            "VpnConnectionId": "vpn-1122334455aabbccd",
            "TransitGatewayId": "tgw-00112233445566aab",
            "Options": {
                "EnableAcceleration": false,
                "StaticRoutesOnly": true,
                "LocalIpv4NetworkCidr": "0.0.0.0/0",
                "RemoteIpv4NetworkCidr": "0.0.0.0/0",
                "TunnelInsideIpVersion": "ipv4"
            },
            "Routes": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "CanadaVPN"
                }
            ],
            "VgwTelemetry": [
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-07-29T10:35:11.000Z",
                    "OutsideIpAddress": "203.0.113.3",
                    "Status": "DOWN",
                    "StatusMessage": ""
                },
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-09-02T09:09:33.000Z",
                    "OutsideIpAddress": "203.0.113.5",
                    "Status": "UP",
                    "StatusMessage": ""
                }
            ]
        }
    ]
}
```


Pour plus d'informations, consultez la section AWS Site-to-Site Fonctionnement du VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

Exemple 2 : Pour décrire vos connexions VPN disponibles

L'`describe-vpn-connections`exemple suivant décrit vos connexions Site-to-Site VPN avec un état de`available`.

```
aws ec2 describe-vpn-connections \
    --filters "Name=state,Values=available"
```

Pour plus d'informations, consultez la section AWS Site-to-Site Fonctionnement du VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

- Pour plus de détails sur l'API, reportez-vous DescribeVpnConnectionsà la section Référence des AWS CLI commandes.

`describe-vpn-gateways`

L'exemple de code suivant montre comment utiliser`describe-vpn-gateways`.

AWS CLI

Pour décrire vos passerelles privées virtuelles

Cet exemple décrit vos passerelles privées virtuelles.

Commande :

```
aws ec2 describe-vpn-gateways
```

Sortie :

```
{
    "VpnGateways": [
        {
            "State": "available",
            "Type": "ipsec.1",
            "VpnGatewayId": "vgw-f211f09b",
            "VpcAttachments": [
                {
                    "State": "attached",
                    "VpcId": "vpc-98eb5ef5"
                }
            ]
        },
```

```
        {
            "State": "available",
            "Type": "ipsec.1",
            "VpnGatewayId": "vgw-9a4cacf3",
            "VpcAttachments": [
                {
                    "State": "attaching",
                    "VpcId": "vpc-a01106c2"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeVpnGatewaysà la section Référence des AWS CLI commandes.

`detach-classic-link-vpc`

L'exemple de code suivant montre comment utiliser`detach-classic-link-vpc`.

AWS CLI

Pour dissocier (détacher) une instance EC2 -Classic d'un VPC

Cet exemple dissocie l'instance i-0598c7d356eba48d7 du VPC vpc-88888888.

Commande :

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-
id vpc-88888888
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous DetachClassicLinkVpcà la section Référence des AWS CLI commandes.

`detach-internet-gateway`

L'exemple de code suivant montre comment utiliser`detach-internet-gateway`.

AWS CLI

Pour détacher une passerelle Internet de votre VPC

L'`detach-internet-gateway`exemple suivant détache la passerelle Internet spécifiée du VPC spécifique.

```
aws ec2 detach-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE \
    --vpc-id vpc-0a60eb65b4EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Passerelles Internet dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DetachInternetGatewayà la section Référence des AWS CLI commandes.

`detach-network-interface`

L'exemple de code suivant montre comment utiliser`detach-network-interface`.

AWS CLI

Pour détacher une interface réseau de votre instance

Cet exemple détache l'interface réseau spécifiée de l'instance spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- Pour plus de détails sur l'API, reportez-vous DetachNetworkInterfaceà la section Référence des AWS CLI commandes.

`detach-verified-access-trust-provider`

L'exemple de code suivant montre comment utiliser`detach-verified-access-trust-provider`.

AWS CLI

Pour détacher un fournisseur de confiance d'une instance

L'`detach-verified-access-trust-provider`exemple suivant détache le fournisseur de confiance Verified Access spécifié de l'instance Verified Access spécifiée.

```
aws ec2 detach-verified-access-trust-provider \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Sortie :

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "Testing Verified Access",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T19:00:38",
        "LastUpdatedTime": "2023-08-25T19:00:38"
    },
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T18:27:56"
    }
}
```

Pour plus d'informations, consultez la section Instances d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous DetachVerifiedAccessTrustProviderà la section Référence des AWS CLI commandes.

`detach-volume`

L'exemple de code suivant montre comment utiliser`detach-volume`.

AWS CLI

Pour détacher un volume d'une instance

Cet exemple de commande détache le volume (`vol-049df61146c4d7901`) de l'instance à laquelle il est attaché.

Commande :

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

Sortie :

```
{
    "AttachTime": "2014-02-27T19:23:06.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "VolumeId": "vol-049df61146c4d7901",
    "State": "detaching",
    "Device": "/dev/sdb"
}
```

- Pour plus de détails sur l'API, reportez-vous DetachVolumeà la section Référence des AWS CLI commandes.

`detach-vpn-gateway`

L'exemple de code suivant montre comment utiliser`detach-vpn-gateway`.

AWS CLI

Pour détacher une passerelle privée virtuelle de votre VPC

Cet exemple détache la passerelle privée virtuelle spécifiée du VPC spécifié. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- Pour plus de détails sur l'API, reportez-vous DetachVpnGatewayà la section Référence des AWS CLI commandes.

`disable-address-transfer`

L'exemple de code suivant montre comment utiliser`disable-address-transfer`.

AWS CLI

Pour désactiver un transfert d'adresse IP élastique

L'`disable-address-transfer`exemple suivant désactive le transfert d'adresse IP élastique pour l'adresse IP élastique spécifiée.

```
aws ec2 disable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf
```

Sortie :

```
{
    "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "AddressTransferStatus": "disabled"
    }
}
```

Pour plus d'informations, consultez Transfer Elastic IP address dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DisableAddressTransferà la section Référence des AWS CLI commandes.

`disable-aws-network-performance-metric-subscription`

L'exemple de code suivant montre comment utiliser`disable-aws-network-performance-metric-subscription`.

AWS CLI

Pour désactiver un abonnement aux métriques

L'`disable-aws-network-performance-metric-subscription`exemple suivant désactive la surveillance de la latence réseau globale entre les régions source et de destination spécifiées.

```
aws ec2 disable-aws-network-performance-metric-subscription \
    --source us-east-1 \
    --destination eu-west-1 \
    --metric aggregate-latency \
    --statistic p50
```

Sortie :

```
{
    "Output": true
}
```

Pour plus d'informations, consultez la section Gérer les CloudWatch abonnements à l'aide de la CLI dans le Guide de l'utilisateur des performances de l'infrastructure.

- Pour plus de détails sur l'API, reportez-vous DisableAwsNetworkPerformanceMetricSubscriptionà la section Référence des AWS CLI commandes.

`disable-ebs-encryption-by-default`

L'exemple de code suivant montre comment utiliser`disable-ebs-encryption-by-default`.

AWS CLI

Pour désactiver le chiffrement EBS par défaut

L'`disable-ebs-encryption-by-default`exemple suivant désactive le chiffrement EBS par défaut pour votre AWS compte dans la région actuelle.

```
aws ec2 disable-ebs-encryption-by-default
```

Sortie :

```
{
    "EbsEncryptionByDefault": false
}
```

- Pour plus de détails sur l'API, reportez-vous DisableEbsEncryptionByDefaultà la section Référence des AWS CLI commandes.

`disable-fast-launch`

L'exemple de code suivant montre comment utiliser`disable-fast-launch`.

AWS CLI

Pour arrêter le lancement rapide d'une image

L'`disable-fast-launch`exemple suivant interrompt le lancement rapide pour l'AMI spécifiée et nettoie les snapshots préprovisionnés existants.

```
aws ec2 disable-fast-launch \
    --image-id ami-01234567890abcedf
```

Sortie :

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
}
```

Pour plus d'informations, consultez Configurer les paramètres de lancement EC2 rapide pour votre AMI Windows dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisableFastLaunchà la section Référence des AWS CLI commandes.

`disable-fast-snapshot-restores`

L'exemple de code suivant montre comment utiliser`disable-fast-snapshot-restores`.

AWS CLI

Pour désactiver la restauration rapide des instantanés

L'`disable-fast-snapshot-restores`exemple suivant désactive la restauration rapide des instantanés pour le cliché spécifié dans la zone de disponibilité spécifiée.

```
aws ec2 disable-fast-snapshot-restores \
    --availability-zones us-east-2a \
    --source-snapshot-ids snap-1234567890abcdef0
```

Sortie :

```
{
    "Successful": [
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2a",
            "State": "disabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.602Z"
        }
    ],
    "Unsuccessful": []
}
```

- Pour plus de détails sur l'API, reportez-vous DisableFastSnapshotRestoresà la section Référence des AWS CLI commandes.

`disable-image-block-public-access`

L'exemple de code suivant montre comment utiliser`disable-image-block-public-access`.

AWS CLI

Pour désactiver le blocage de l'accès public AMIs dans la région spécifiée

L'`disable-image-block-public-access`exemple suivant désactive le blocage de l'accès public AMIs au niveau du compte dans la région spécifiée.

```
aws ec2 disable-image-block-public-access \
    --region us-east-1
```

Sortie :

```
{
    "ImageBlockPublicAccessState": "unblocked"
}
```

Pour plus d'informations, consultez Bloquer l'accès public à votre compte AMIs dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisableImageBlockPublicAccessà la section Référence des AWS CLI commandes.

`disable-image-deprecation`

L'exemple de code suivant montre comment utiliser`disable-image-deprecation`.

AWS CLI

Pour annuler la dépréciation d'une AMI

L'`disable-image-deprecation`exemple suivant annule la dépréciation d'une AMI, qui supprime le `DeprecationTime` champ de la sortie. `describe-images` Pour ce faire, vous devez être le propriétaire de l'AMI.

```
aws ec2 disable-image-deprecation \
    --image-id ami-1234567890abcdef0
```

Sortie :

```
{
    "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",
    "Return": "true"
}
```

Pour plus d'informations, consultez la section Dépréciation d'une AMI dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisableImageDeprecationà la section Référence des AWS CLI commandes.

`disable-image-deregistration-protection`

L'exemple de code suivant montre comment utiliser`disable-image-deregistration-protection`.

AWS CLI

Pour désactiver la protection contre la désinscription

L'`disable-image-deregistration-protection`exemple suivant désactive la protection contre le désenregistrement pour l'image spécifiée.

```
aws ec2 disable-image-deregistration-protection \
    --image-id ami-0b1a928a144a74ec9
```

Sortie :

```
{
    "Return": "disabled"
}
```

Pour plus d'informations, consultez la section Protéger une AMI contre la désinscription dans le guide de l'utilisateur Amazon EC2 .

- Pour plus de détails sur l'API, reportez-vous DisableImageDeregistrationProtectionà la section Référence des AWS CLI commandes.

`disable-image`

L'exemple de code suivant montre comment utiliser`disable-image`.

AWS CLI

Pour désactiver une AMI

L'`disable-image`exemple suivant désactive l'AMI spécifiée.

```
aws ec2 disable-image \
    --image-id ami-1234567890abcdef0
```

Sortie :

```
{
    "Return": "true"
}
```

Pour plus d'informations, consultez la section Désactiver une AMI dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisableImageà la section Référence des AWS CLI commandes.

`disable-ipam-organization-admin-account`

L'exemple de code suivant montre comment utiliser`disable-ipam-organization-admin-account`.

AWS CLI

Pour désactiver l'administrateur IPAM délégué

Dans certains scénarios, vous allez intégrer IPAM à AWS Organizations. Dans ce cas, le compte de gestion des AWS Organisations délègue un compte membre d' AWS Organizations en tant qu'administrateur IPAM.

Dans cet exemple, vous êtes le compte de gestion des AWS Organizations qui a délégué le compte administrateur IPAM et vous souhaitez empêcher ce compte d'être l'administrateur IPAM.

Vous pouvez utiliser n'importe quelle AWS région `--region` lorsque vous faites cette demande. Vous n'êtes pas obligé d'utiliser la région dans laquelle vous avez initialement délégué l'administrateur, où l'IPAM a été créé, ni une région d'exploitation IPAM. Si vous désactivez le compte d'administrateur délégué, vous pouvez le réactiver à tout moment ou déléguer un nouveau compte en tant qu'administrateur IPAM.

L'`disable-ipam-organization-admin-account`exemple suivant désactive l'administrateur IPAM délégué dans votre AWS compte.

```
aws ec2 disable-ipam-organization-admin-account \
    --delegated-admin-account-id 320805250157 \
    --region ap-south-1
```

Sortie :

```
{
    "Success": true
}
```

Pour plus d'informations, consultez la section Intégrer l'IPAM aux comptes d'une AWS organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DisableIpamOrganizationAdminAccountà la section Référence des AWS CLI commandes.

`disable-serial-console-access`

L'exemple de code suivant montre comment utiliser`disable-serial-console-access`.

AWS CLI

Pour désactiver l'accès à la console EC2 série pour votre compte

L'`disable-serial-console-access`exemple suivant désactive l'accès du compte à la console série.

```
aws ec2 disable-serial-console-access
```

Sortie :

```
{
    "SerialConsoleAccessEnabled": false
}
```

Pour plus d'informations, consultez EC2 Serial Console dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisableSerialConsoleAccessà la section Référence des AWS CLI commandes.

`disable-snapshot-block-public-access`

L'exemple de code suivant montre comment utiliser`disable-snapshot-block-public-access`.

AWS CLI

Pour désactiver le blocage de l'accès public aux instantanés

L'`disable-snapshot-block-public-access`exemple suivant désactive le blocage de l'accès public aux instantanés afin de permettre le partage public de vos instantanés.

```
aws ec2 disable-snapshot-block-public-access
```

Sortie :

```
{
    "State": "unblocked"
}
```

Pour plus d'informations, consultez Bloquer l'accès public aux instantanés dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous DisableSnapshotBlockPublicAccessà la section Référence des AWS CLI commandes.

`disable-transit-gateway-route-table-propagation`

L'exemple de code suivant montre comment utiliser`disable-transit-gateway-route-table-propagation`.

AWS CLI

Pour désactiver une pièce jointe à une passerelle de transit afin de propager les itinéraires vers la table de routage de propagation spécifiée

L'`disable-transit-gateway-route-table-propagation`exemple suivant désactive la pièce jointe spécifiée pour propager les itinéraires vers la table de routage de propagation spécifiée.

```
aws ec2 disable-transit-gateway-route-table-propagation \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
```

```
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Sortie :

```
{
    "Propagation": {
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
        "State": "disabled"
    }
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DisableTransitGatewayRouteTablePropagationà la section Référence des AWS CLI commandes.

`disable-vgw-route-propagation`

L'exemple de code suivant montre comment utiliser`disable-vgw-route-propagation`.

AWS CLI

Pour désactiver la propagation des itinéraires

Cet exemple empêche la passerelle privée virtuelle spécifiée de propager des itinéraires statiques vers la table de routage spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-
id vgw-9a4cacf3
```

- Pour plus de détails sur l'API, reportez-vous DisableVgwRoutePropagationà la section Référence des AWS CLI commandes.

`disable-vpc-classic-link-dns-support`

L'exemple de code suivant montre comment utiliser`disable-vpc-classic-link-dns-support`.

AWS CLI

Pour désactiver le support ClassicLink DNS pour un VPC

Cet exemple désactive le support ClassicLink DNS pour`vpc-88888888`.

Commande :

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous DisableVpcClassicLinkDnsSupportà la section Référence des AWS CLI commandes.

`disable-vpc-classic-link`

L'exemple de code suivant montre comment utiliser`disable-vpc-classic-link`.

AWS CLI

Pour désactiver ClassicLink pour un VPC

Cet exemple désactive ClassicLink vpc-8888888.

Commande :

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous DisableVpcClassicLinkà la section Référence des AWS CLI commandes.

`disassociate-address`

L'exemple de code suivant montre comment utiliser`disassociate-address`.

AWS CLI

Pour dissocier une adresse IP élastique dans -Classic EC2

Cet exemple dissocie une adresse IP élastique d'une instance dans EC2 -Classic. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

Pour dissocier une adresse IP élastique dans -VPC EC2

Cet exemple dissocie une adresse IP Elastic d'une instance dans un VPC. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- Pour plus de détails sur l'API, reportez-vous DisassociateAddressà la section Référence des AWS CLI commandes.

`disassociate-client-vpn-target-network`

L'exemple de code suivant montre comment utiliser`disassociate-client-vpn-target-network`.

AWS CLI

Pour dissocier un réseau d'un point de terminaison VPN client

L'`disassociate-client-vpn-target-network`exemple suivant dissocie le réseau cible associé à l'ID d'`cvpn-assoc-12312312312312312`association pour le point de terminaison VPN Client spécifié.

```
aws ec2 disassociate-client-vpn-target-network \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
```

```
    --association-id cvpn-assoc-12312312312312312
```

Sortie :

```
{
    "AssociationId": "cvpn-assoc-12312312312312312",
    "Status": {
        "Code": "disassociating"
    }
}
```

Pour plus d'informations, consultez la section Réseaux cibles dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous DisassociateClientVpnTargetNetworkà la section Référence des AWS CLI commandes.

`disassociate-iam-instance-profile`

L'exemple de code suivant montre comment utiliser`disassociate-iam-instance-profile`.

AWS CLI

Pour dissocier un profil d'instance IAM

Cet exemple dissocie un profil d'instance IAM de l'ID d'association. `iip-assoc-05020b59952902f5f`

Commande :

```
aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-05020b59952902f5f
```

Sortie :

```
{
  "IamInstanceProfileAssociation": {
      "InstanceId": "i-123456789abcde123",
      "State": "disassociating",
      "AssociationId": "iip-assoc-05020b59952902f5f",
      "IamInstanceProfile": {
          "Id": "AIPAI5IVIHMFFYY2DKV5Y",
```

```
            "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateIamInstanceProfileà la section Référence des AWS CLI commandes.

`disassociate-instance-event-window`

L'exemple de code suivant montre comment utiliser`disassociate-instance-event-window`.

AWS CLI

Exemple 1 : pour dissocier une ou plusieurs instances d'une fenêtre d'événements

L'`disassociate-instance-event-window`exemple suivant dissocie une ou plusieurs instances d'une fenêtre d'événements. Spécifiez le `instance-event-window-id` paramètre pour définir la fenêtre d'événements. Pour dissocier les instances, spécifiez le `association-target` paramètre, et pour les valeurs des paramètres, spécifiez une ou plusieurs instances IDs.

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour dissocier les balises d'instance d'une fenêtre d'événements

L'`disassociate-instance-event-window`exemple suivant dissocie les balises d'instance d'une fenêtre d'événements. Spécifiez le `instance-event-window-id` paramètre pour définir la fenêtre d'événements. Pour dissocier des étiquettes d'instance, spécifiez le paramètre `association-target`, et pour les valeurs de paramètre, spécifiez une ou plusieurs étiquettes.

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour dissocier un hôte dédié d'une fenêtre d'événements

L'`disassociate-instance-event-window`exemple suivant dissocie un hôte dédié d'une fenêtre d'événements. Spécifiez le `instance-event-window-id` paramètre pour définir la fenêtre d'événements. Pour dissocier un hôte dédié, spécifiez le `association-target` paramètre, et pour les valeurs des paramètres, spécifiez un ou plusieurs hôtes IDs dédiés.

```
aws ec2 disassociate-instance-event-window \
```

```
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DisassociateInstanceEventWindowà la section Référence des AWS CLI commandes.

`disassociate-ipam-resource-discovery`

L'exemple de code suivant montre comment utiliser`disassociate-ipam-resource-discovery`.

AWS CLI

Pour dissocier une découverte de ressources d'un IPAM

Dans cet exemple, vous êtes un compte d'administrateur délégué IPAM et vous souhaitez dissocier une découverte de ressources IPAM de votre IPAM. Vous avez exécuté la commande describe et vous avez remarqué que vous vouliez la `"ResourceDiscoveryStatus": "not-found"` dissocier de votre IPAM pour faire de la place à d'autres associations.

L'`disassociate-ipam-resource-discovery`exemple suivant dissocie une découverte de ressources IPAM dans votre AWS compte.

```
aws ec2 disassociate-ipam-resource-discovery \
    --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82
 \
    --region us-east-1
```

Sortie :

```
{
    "IpamResourceDiscoveryAssociation": {
        "OwnerId": "320805250157",
        "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
        "IpamResourceDiscoveryAssociationArn":
 "arn:aws:ec2::320805250157:ipam-resource-discovery-association/ipam-res-disco-
assoc-04382a6346357cf82",
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
        "IpamId": "ipam-005f921c17ebd5107",
        "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
        "IpamRegion": "us-east-1",
        "IsDefault": false,
        "ResourceDiscoveryStatus": "not-found",
        "State": "disassociate-in-progress"
    }
}
```

Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous DisassociateIpamResourceDiscoveryà la section Référence des AWS CLI commandes.

`disassociate-nat-gateway-address`

L'exemple de code suivant montre comment utiliser`disassociate-nat-gateway-address`.

AWS CLI

Pour dissocier une adresse IP élastique d'une passerelle NAT publique

L'`disassociate-nat-gateway-address`exemple suivant dissocie l'adresse IP élastique spécifiée de la passerelle NAT publique spécifiée.

```
aws ec2 disassociate-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --association-ids eipassoc-0f96bdca17EXAMPLE
```

Sortie :

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
            "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
            "PrivateIp": "10.0.0.74",
            "PublicIp": "3.211.231.218",
            "AssociationId": "eipassoc-0f96bdca17EXAMPLE",
            "IsPrimary": false,
            "Status": "disassociating"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DisassociateNatGatewayAddressà la section Référence des AWS CLI commandes.

`disassociate-route-table`

L'exemple de code suivant montre comment utiliser`disassociate-route-table`.

AWS CLI

Pour dissocier une table de routage

Cet exemple dissocie la table de routage spécifiée du sous-réseau spécifié. Si la commande réussit, aucune sortie n’est renvoyée.

Commande :

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- Pour plus de détails sur l'API, reportez-vous DisassociateRouteTableà la section Référence des AWS CLI commandes.

`disassociate-security-group-vpc`

L'exemple de code suivant montre comment utiliser`disassociate-security-group-vpc`.

AWS CLI

Pour dissocier un groupe de sécurité d'un VPC

L'`disassociate-security-group-vpc`exemple suivant dissocie le groupe de sécurité spécifié du VPC spécifié.

```
aws ec2 disassociate-security-group-vpc \
    --group-id sg-04dbb43907d3f8a78 \
    --vpc-id vpc-0bf4c2739bc05a694
```

Sortie :

```
{
    "State": "disassociating"
}
```

Pour plus d'informations, consultez Associer des groupes de sécurité à plusieurs VPCs dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous DisassociateSecurityGroupVpcà la section Référence des AWS CLI commandes.

`disassociate-subnet-cidr-block`

L'exemple de code suivant montre comment utiliser`disassociate-subnet-cidr-block`.

AWS CLI

Pour dissocier un bloc IPv6 CIDR d'un sous-réseau

Cet exemple dissocie un bloc d'adresse IPv6 CIDR d'un sous-réseau à l'aide de l'ID d'association du bloc d'adresse CIDR.

Commande :

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

Sortie :

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
      "Ipv6CidrBlockState": {
          "State": "disassociating"
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateSubnetCidrBlockà la section Référence des AWS CLI commandes.

`disassociate-transit-gateway-multicast-domain`

L'exemple de code suivant montre comment utiliser`disassociate-transit-gateway-multicast-domain`.

AWS CLI

Pour dissocier les sous-réseaux d'un domaine de multidiffusion

L'`disassociate-transit-gateway-multicast-domain`exemple suivant dissocie un sous-réseau du domaine de multidiffusion spécifié.

```
aws ec2 disassociate-transit-gateway-multicast-domain \
    --transit-gateway-attachment-id tgw-attach-070e571cd1EXAMPLE \
    --subnet-id subnet-000de86e3bEXAMPLE \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Sortie :

```
{
    "Associations": {
```

```
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
        "ResourceId": "vpc-7EXAMPLE",
        "ResourceType": "vpc",
        "Subnets": [
            {
                "SubnetId": "subnet-000de86e3bEXAMPLE",
                "State": "disassociating"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Domaines de multidiffusion dans le guide des passerelles de transit '.

- Pour plus de détails sur l'API, reportez-vous DisassociateTransitGatewayMulticastDomainà la section Référence des AWS CLI commandes.

`disassociate-transit-gateway-route-table`

L'exemple de code suivant montre comment utiliser`disassociate-transit-gateway-route-table`.

AWS CLI

Pour dissocier une table de routage d'une passerelle de transit d'une ressource attachée

L'`disassociate-transit-gateway-route-table`exemple suivant dissocie la pièce jointe spécifiée de la table de routage de la passerelle de transit.

```
aws ec2 disassociate-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE
```

Sortie :

```
{
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
        "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
```

```
        "ResourceType": "direct-connect-gateway",
        "State": "disassociating"
    }
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous DisassociateTransitGatewayRouteTableà la section Référence des AWS CLI commandes.

`disassociate-vpc-cidr-block`

L'exemple de code suivant montre comment utiliser`disassociate-vpc-cidr-block`.

AWS CLI

Pour dissocier un bloc IPv6 CIDR d'un VPC

Cet exemple dissocie un bloc d'adresse IPv6 CIDR d'un VPC à l'aide de l'ID d'association du bloc d'adresse CIDR.

Commande :

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

Sortie :

```
{
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
      "AssociationId": "vpc-cidr-assoc-eca54085",
      "Ipv6CidrBlockState": {
          "State": "disassociating"
      }
  },
  "VpcId": "vpc-a034d6c4"
}
```

Pour dissocier un bloc IPv4 CIDR d'un VPC

Cet exemple dissocie un bloc IPv4 CIDR d'un VPC.

Commande :

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

Sortie :

```
{
  "CidrBlockAssociation": {
      "AssociationId": "vpc-cidr-assoc-0287ac6b",
      "CidrBlock": "172.18.0.0/16",
      "CidrBlockState": {
          "State": "disassociating"
      }
  },
  "VpcId": "vpc-27621243"
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateVpcCidrBlockà la section Référence des AWS CLI commandes.

`enable-address-transfer`

L'exemple de code suivant montre comment utiliser`enable-address-transfer`.

AWS CLI

Pour activer un transfert d'adresse IP élastique

L'`enable-address-transfer`exemple suivant active le transfert d'adresse IP élastique pour l'adresse IP élastique spécifiée vers le compte spécifié.

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

Sortie :

```
{
     "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
```

```
        "TransferAccountId": "123456789012",
        "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",
        "AddressTransferStatus": "pending"
    }
}
```

Pour plus d'informations, consultez Transfer Elastic IP address dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous EnableAddressTransferà la section Référence des AWS CLI commandes.

`enable-aws-network-performance-metric-subscription`

L'exemple de code suivant montre comment utiliser`enable-aws-network-performance-metric-subscription`.

AWS CLI

Pour activer un abonnement aux métriques

L'`enable-aws-network-performance-metric-subscription`exemple suivant permet de surveiller la latence globale du réseau entre les régions source et de destination spécifiées.

```
aws ec2 enable-aws-network-performance-metric-subscription \
    --source us-east-1 \
    --destination eu-west-1 \
    --metric aggregate-latency \
    --statistic p50
```

Sortie :

```
{
    "Output": true
}
```

Pour plus d'informations, consultez la section Gérer les abonnements dans le Guide de l'utilisateur sur les performances de l'infrastructure.

- Pour plus de détails sur l'API, reportez-vous EnableAwsNetworkPerformanceMetricSubscriptionà la section Référence des AWS CLI commandes.

`enable-ebs-encryption-by-default`

L'exemple de code suivant montre comment utiliser`enable-ebs-encryption-by-default`.

AWS CLI

Pour activer le chiffrement EBS par défaut

L'`enable-ebs-encryption-by-default`exemple suivant active le chiffrement EBS par défaut pour votre AWS compte dans la région actuelle.

```
aws ec2 enable-ebs-encryption-by-default
```

Sortie :

```
{
    "EbsEncryptionByDefault": true
}
```

- Pour plus de détails sur l'API, reportez-vous EnableEbsEncryptionByDefaultà la section Référence des AWS CLI commandes.

`enable-fast-launch`

L'exemple de code suivant montre comment utiliser`enable-fast-launch`.

AWS CLI

Pour démarrer le lancement rapide d'une image

L'`enable-fast-launch`exemple suivant configure l'AMI spécifiée pour Fast Launch et définit le nombre maximum d'instances parallèles à lancer à 6. Le type de ressource à utiliser pour approvisionner préalablement l'AMI est défini sur `snapshot`, qui est également la valeur par défaut.

```
aws ec2 enable-fast-launch \
    --image-id ami-01234567890abcedf \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

Sortie :

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {
        "TargetResourceCount": 10
    },
    "LaunchTemplate": {},
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "enabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

Pour plus d'informations, consultez Configurer les paramètres de lancement EC2 rapide pour votre AMI Windows dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous EnableFastLaunchà la section Référence des AWS CLI commandes.

`enable-fast-snapshot-restores`

L'exemple de code suivant montre comment utiliser`enable-fast-snapshot-restores`.

AWS CLI

Pour activer la restauration rapide des instantanés

L'`enable-fast-snapshot-restores`exemple suivant permet de restaurer rapidement un instantané pour le cliché spécifié dans les zones de disponibilité spécifiées.

```
aws ec2 enable-fast-snapshot-restores \
    --availability-zones us-east-2a us-east-2b \
    --source-snapshot-ids snap-1234567890abcdef0
```

Sortie :

```
{
    "Successful": [
```

```
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2a",
            "State": "enabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.602Z"
        },
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2b",
            "State": "enabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z"
        }
    ],
    "Unsuccessful": []
}
```

- Pour plus de détails sur l'API, reportez-vous EnableFastSnapshotRestoresà la section Référence des AWS CLI commandes.

`enable-image-block-public-access`

L'exemple de code suivant montre comment utiliser`enable-image-block-public-access`.

AWS CLI

Pour activer le blocage de l'accès public AMIs dans la région spécifiée

L'`enable-image-block-public-access`exemple suivant permet de bloquer l'accès public AMIs au niveau du compte dans la région spécifiée.

```
aws ec2 enable-image-block-public-access \
    --region us-east-1 \
    --image-block-public-access-state block-new-sharing
```

Sortie :

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
```

```
}
```

Pour plus d'informations, consultez Bloquer l'accès public à votre compte AMIs dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous EnableImageBlockPublicAccessà la section Référence des AWS CLI commandes.

`enable-image-deprecation`

L'exemple de code suivant montre comment utiliser`enable-image-deprecation`.

AWS CLI

Pour déprécier une AMI

L'`enable-image-deprecation`exemple suivant rend obsolète une AMI à une date et à une heure spécifiques. Si vous spécifiez une valeur pour les secondes, Amazon EC2 arrondit les secondes à la minute la plus proche. Pour ce faire, vous devez être le propriétaire de l'AMI.

```
aws ec2 enable-image-deprecation \
    --image-id ami-1234567890abcdef0 \
    --deprecate-at '2022-10-15T13:17:12.000Z'
```

Sortie :

```
{
    "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",
    "Return": "true"
}
```

Pour plus d'informations, consultez la section Dépréciation d'une AMI dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous EnableImageDeprecationà la section Référence des AWS CLI commandes.

`enable-image-deregistration-protection`

L'exemple de code suivant montre comment utiliser`enable-image-deregistration-protection`.

AWS CLI

Pour activer la protection contre la désinscription

L'`enable-image-deregistration-protection`exemple suivant active la protection contre le désenregistrement pour l'image spécifiée.

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0b1a928a144a74ec9
```

Sortie :

```
{
    "Return": "enabled-without-cooldown"
}
```

Pour plus d'informations, consultez la section Protéger une EC2 AMI contre la désinscription dans le guide de l'utilisateur Amazon EC2 .

- Pour plus de détails sur l'API, reportez-vous EnableImageDeregistrationProtectionà la section Référence des AWS CLI commandes.

`enable-image`

L'exemple de code suivant montre comment utiliser`enable-image`.

AWS CLI

Pour activer une AMI

L'`enable-image`exemple suivant active l'AMI spécifiée.

```
aws ec2 enable-image \
    --image-id ami-1234567890abcdef0
```

Sortie :

```
{
    "Return": "true"
}
```

Pour plus d'informations, consultez la section Désactiver une AMI dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous EnableImageà la section Référence des AWS CLI commandes.

`enable-ipam-organization-admin-account`

L'exemple de code suivant montre comment utiliser`enable-ipam-organization-admin-account`.

AWS CLI

Pour intégrer les AWS Organizations et déléguer un compte membre en tant que compte IPAM

L'`enable-ipam-organization-admin-account`exemple suivant intègre IPAM à AWS Organizations et délègue un compte membre en tant que compte IPAM.

```
aws ec2 enable-ipam-organization-admin-account \
    --delegated-admin-account-id 320805250157
```

Sortie :

```
{
    "Success": true
}
```

Pour plus d'informations, consultez Intégrer l'IPAM aux AWS organisations dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous EnableIpamOrganizationAdminAccountà la section Référence des AWS CLI commandes.

`enable-reachability-analyzer-organization-sharing`

L'exemple de code suivant montre comment utiliser`enable-reachability-analyzer-organization-sharing`.

AWS CLI

Pour permettre un accès fiable à Reachability Analyzer

L'`enable-reachability-analyzer-organization-sharing`exemple suivant active un accès sécurisé pour Reachability Analyzer.

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Analyses entre comptes dans le guide de l'utilisateur de Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous EnableReachabilityAnalyzerOrganizationSharingà la section Référence des AWS CLI commandes.

`enable-serial-console-access`

L'exemple de code suivant montre comment utiliser`enable-serial-console-access`.

AWS CLI

Pour activer l'accès à la console série pour votre compte

L'`enable-serial-console-access`exemple suivant active l'accès du compte à la console série.

```
aws ec2 enable-serial-console-access
```

Sortie :

```
{
    "SerialConsoleAccessEnabled": true
}
```

Pour plus d'informations, consultez EC2 Serial Console dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous EnableSerialConsoleAccessà la section Référence des AWS CLI commandes.

`enable-snapshot-block-public-access`

L'exemple de code suivant montre comment utiliser`enable-snapshot-block-public-access`.

AWS CLI

Pour activer le blocage de l'accès public aux instantanés

L'`enable-snapshot-block-public-access`exemple suivant bloque tout partage public de vos instantanés.

```
aws ec2 enable-snapshot-block-public-access \
    --state block-all-sharing
```

Sortie :

```
{
    "State": "block-all-sharing"
}
```

Pour plus d'informations, consultez Bloquer l'accès public aux instantanés dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous EnableSnapshotBlockPublicAccessà la section Référence des AWS CLI commandes.

`enable-transit-gateway-route-table-propagation`

L'exemple de code suivant montre comment utiliser`enable-transit-gateway-route-table-propagation`.

AWS CLI

Pour permettre à une pièce jointe à une passerelle de transit de propager les itinéraires vers la table de routage de propagation spécifiée

L'`enable-transit-gateway-route-table-propagation`exemple suivant permet à la pièce jointe spécifiée de propager des itinéraires vers la table de routage de propagation spécifiée.

```
aws ec2 enable-transit-gateway-route-table-propagation \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Sortie :

```
{
    "Propagation": {
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
        "State": "disabled"
    }
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous EnableTransitGatewayRouteTablePropagationà la section Référence des AWS CLI commandes.

`enable-vgw-route-propagation`

L'exemple de code suivant montre comment utiliser`enable-vgw-route-propagation`.

AWS CLI

Pour activer la propagation des itinéraires

Cet exemple permet à la passerelle privée virtuelle spécifiée de propager des itinéraires statiques vers la table de routage spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-
id vgw-9a4cacf3
```

- Pour plus de détails sur l'API, reportez-vous EnableVgwRoutePropagationà la section Référence des AWS CLI commandes.

`enable-volume-io`

L'exemple de code suivant montre comment utiliser`enable-volume-io`.

AWS CLI

I/O Pour activer un volume

Cet exemple active I/O le mode « on volume `vol-1234567890abcdef0` ».

Commande :

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous EnableVolumeIoà la section Référence des AWS CLI commandes.

`enable-vpc-classic-link-dns-support`

L'exemple de code suivant montre comment utiliser`enable-vpc-classic-link-dns-support`.

AWS CLI

Pour activer le support ClassicLink DNS pour un VPC

Cet exemple active le support ClassicLink DNS pour`vpc-88888888`.

Commande :

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous EnableVpcClassicLinkDnsSupportà la section Référence des AWS CLI commandes.

`enable-vpc-classic-link`

L'exemple de code suivant montre comment utiliser`enable-vpc-classic-link`.

AWS CLI

Pour activer un VPC pour ClassicLink

Cet exemple active vpc-8888888 pour. ClassicLink

Commande :

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous EnableVpcClassicLinkà la section Référence des AWS CLI commandes.

`export-client-vpn-client-certificate-revocation-list`

L'exemple de code suivant montre comment utiliser`export-client-vpn-client-certificate-revocation-list`.

AWS CLI

Pour exporter une liste de révocation de certificats clients

L'`export-client-vpn-client-certificate-revocation-list`exemple suivant exporte la liste de révocation des certificats clients pour le point de terminaison VPN client spécifié. Dans cet exemple, le résultat est renvoyé au format texte pour en faciliter la lecture.

```
aws ec2 export-client-vpn-client-certificate-revocation-list \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --output text
```

Sortie :

```
-----BEGIN X509 CRL-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
```

```
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END X509 CRL-----
STATUS      pending
```

Pour plus d'informations, consultez les listes de révocation des certificats clients dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous ExportClientVpnClientCertificateRevocationListà la section Référence des AWS CLI commandes.

`export-client-vpn-client-configuration`

L'exemple de code suivant montre comment utiliser`export-client-vpn-client-configuration`.

AWS CLI

Pour exporter la configuration du client

L'`export-client-vpn-client-configuration`exemple suivant exporte la configuration du client pour le point de terminaison VPN client spécifié. Dans cet exemple, le résultat est renvoyé au format texte pour en faciliter la lecture.

```
aws ec2 export-client-vpn-client-configuration \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --output text
```

Sortie :

```
client
dev tun
```

```
proto udp
remote cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-south-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
cipher AES-256-GCM
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----
</ca>
reneg-sec 0
```

Pour plus d'informations, consultez la section Exportation du fichier de configuration du point de terminaison du VPN AWS client dans le Guide de l'administrateur du VPN client.

- Pour plus de détails sur l'API, reportez-vous ExportClientVpnClientConfigurationà la section Référence des AWS CLI commandes.

`export-image`

L'exemple de code suivant montre comment utiliser`export-image`.

AWS CLI

Pour exporter une machine virtuelle depuis une AMI

L'`export-image`exemple suivant exporte l'AMI spécifiée vers le compartiment spécifié dans le format spécifié.

```
aws ec2 export-image \
    --image-id ami-1234567890abcdef0 \
    --disk-image-format VMDK \
    --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/
```

Sortie :

```
{
    "DiskImageFormat": "vmdk",
    "ExportImageTaskId": "export-ami-1234567890abcdef0"
    "ImageId": "ami-1234567890abcdef0",
    "RoleName": "vmimport",
    "Progress": "0",
    "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
    },
    "Status": "active",
    "StatusMessage": "validating"
}
```

- Pour plus de détails sur l'API, reportez-vous ExportImageà la section Référence des AWS CLI commandes.

`get-associated-ipv6-pool-cidrs`

L'exemple de code suivant montre comment utiliser`get-associated-ipv6-pool-cidrs`.

AWS CLI

Pour obtenir les associations d'un pool d' IPv6 adresses

L'`get-associated-ipv6-pool-cidrs`exemple suivant obtient les associations pour le pool d' IPv6 adresses spécifié.

```
aws ec2 get-associated-ipv6-pool-cidrs \
    --pool-id ipv6pool-ec2-012345abc12345abc
```

Sortie :

```
{
    "Ipv6CidrAssociations": [
        {
            "Ipv6Cidr": "2001:db8:1234:1a00::/56",
            "AssociatedResource": "vpc-111111222222333ab"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous à la section GetAssociatedIpv6 PoolCidrs de AWS CLI la section Référence des commandes.

`get-aws-network-performance-data`

L'exemple de code suivant montre comment utiliser`get-aws-network-performance-data`.

AWS CLI

Pour obtenir des données sur les performances du réseau

L'`get-aws-network-performance-data`exemple suivant extrait des données relatives aux performances du réseau entre les régions spécifiées au cours de la période spécifiée.

```
aws ec2 get-aws-network-performance-data \
    --start-time 2022-10-26T12:00:00.000Z \
    --end-time 2022-10-26T12:30:00.000Z \
    --data-queries Id=my-query,Source=us-east-1,Destination=eu-
west-1,Metric=aggregate-latency,Statistic=p50,Period=five-minutes
```

Sortie :

```
{
    "DataResponses": [
        {
            "Id": "my-query",
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes",
```

```
            "MetricPoints": [
                {
                    "StartDate": "2022-10-26T12:00:00+00:00",
                    "EndDate": "2022-10-26T12:05:00+00:00",
                    "Value": 62.44349,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:05:00+00:00",
                    "EndDate": "2022-10-26T12:10:00+00:00",
                    "Value": 62.483498,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:10:00+00:00",
                    "EndDate": "2022-10-26T12:15:00+00:00",
                    "Value": 62.51248,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:15:00+00:00",
                    "EndDate": "2022-10-26T12:20:00+00:00",
                    "Value": 62.635475,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:20:00+00:00",
                    "EndDate": "2022-10-26T12:25:00+00:00",
                    "Value": 62.733974,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:25:00+00:00",
                    "EndDate": "2022-10-26T12:30:00+00:00",
                    "Value": 62.773975,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:30:00+00:00",
                    "EndDate": "2022-10-26T12:35:00+00:00",
                    "Value": 62.75349,
                    "Status": "OK"
                }
            ]
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveiller les performances du réseau dans le Guide de l'utilisateur des performances de l'infrastructure.

- Pour plus de détails sur l'API, reportez-vous GetAwsNetworkPerformanceDataà la section Référence des AWS CLI commandes.

`get-capacity-reservation-usage`

L'exemple de code suivant montre comment utiliser`get-capacity-reservation-usage`.

AWS CLI

Pour consulter l'utilisation des réservations de capacité sur l'ensemble AWS des comptes

L'`get-capacity-reservation-usage`exemple suivant affiche les informations d'utilisation pour la réservation de capacité spécifiée.

```
aws ec2 get-capacity-reservation-usage \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Sortie :

```
{
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "InstanceUsages": [
        {
            "UsedInstanceCount": 1,
            "AccountId": "123456789012"
        }
    ],
    "AvailableInstanceCount": 4,
    "TotalInstanceCount": 5,
    "State": "active",
    "InstanceType": "t2.medium"
}
```

Pour plus d'informations, consultez la section sur les réservations de capacité partagée dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetCapacityReservationUsageà la section Référence des AWS CLI commandes.

`get-coip-pool-usage`

L'exemple de code suivant montre comment utiliser`get-coip-pool-usage`.

AWS CLI

Pour obtenir l'utilisation du pool d'adresses IP appartenant au client

L'`get-coip-pool-usage`exemple suivant permet d'obtenir les détails d'utilisation du pool d'adresses IP appartenant au client spécifié.

```
aws ec2 get-coip-pool-usage \
    --pool-id ipv4pool-coip-123a45678bEXAMPLE
```

Sortie :

```
{
    "CoipPoolId": "ipv4pool-coip-123a45678bEXAMPLE",
    "CoipAddressUsages": [
        {
            "CoIp": "0.0.0.0"
        },
        {
            "AllocationId": "eipalloc-123ab45c6dEXAMPLE",
            "AwsAccountId": "123456789012",
            "CoIp": "0.0.0.0"
        },
        {
            "AllocationId": "eipalloc-123ab45c6dEXAMPLE",
            "AwsAccountId": "123456789111",
            "CoIp": "0.0.0.0"
        }
    ],
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
}
```

Pour plus d'informations, consultez la section Adresses IP appartenant aux clients dans le Guide de l'utilisateur d'AWS Outposts pour les racks Outposts.

- Pour plus de détails sur l'API, reportez-vous GetCoipPoolUsageà la section Référence des AWS CLI commandes.

`get-console-output`

L'exemple de code suivant montre comment utiliser`get-console-output`.

AWS CLI

Exemple 1 : pour obtenir la sortie de la console

L'`get-console-output`exemple suivant obtient la sortie de console pour l'instance Linux spécifiée.

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0
```

Sortie :

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-07-25T21:23:53.000Z",
    "Output": "..."
}
```

Pour plus d'informations, consultez la sortie de la console Instance dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour obtenir la dernière sortie de console

L'`get-console-output`exemple suivant obtient la dernière sortie de console pour l'instance Linux spécifiée.

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0 \
    --latest \
    --output text
```

Sortie :

```
i-1234567890abcdef0 [    0.000000] Command line: root=LABEL=/ console=tty1
 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point
 registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
...
Cloud-init v. 0.7.6 finished at Wed, 09 May 2018 19:01:13 +0000. Datasource
 DataSourceEc2.  Up 21.50 seconds
Amazon Linux AMI release 2018.03
Kernel 4.14.26-46.32.amzn1.x
```

Pour plus d'informations, consultez la sortie de la console Instance dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetConsoleOutputà la section Référence des AWS CLI commandes.

`get-console-screenshot`

L'exemple de code suivant montre comment utiliser`get-console-screenshot`.

AWS CLI

Pour récupérer une capture d'écran d'une instance en cours d'exécution

L'`get-console-screenshot`exemple suivant récupère une capture d'écran de l'instance spécifiée au format .jpg. La capture d'écran est renvoyée sous forme de chaîne codée en Base64.

```
aws ec2 get-console-screenshot \
    --instance-id i-1234567890abcdef0
```

Sortie :

```
{
    "ImageData": "997987/8kgj49ikjhewkwwe0008084EXAMPLE",
    "InstanceId": "i-1234567890abcdef0"
}
```

- Pour plus de détails sur l'API, reportez-vous GetConsoleScreenshotà la section Référence des AWS CLI commandes.

`get-default-credit-specification`

L'exemple de code suivant montre comment utiliser`get-default-credit-specification`.

AWS CLI

Pour décrire l'option de crédit par défaut

L'`get-default-credit-specification`exemple suivant décrit l'option de crédit par défaut pour les instances T2.

```
aws ec2 get-default-credit-specification \
    --instance-family t2
```

Sortie :

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "standard"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDefaultCreditSpecificationà la section Référence des AWS CLI commandes.

`get-ebs-default-kms-key-id`

L'exemple de code suivant montre comment utiliser`get-ebs-default-kms-key-id`.

AWS CLI

Pour décrire votre clé CMK par défaut pour le chiffrement EBS

L'`get-ebs-default-kms-key-id`exemple suivant décrit la clé CMK par défaut pour le chiffrement EBS de votre AWS compte.

```
aws ec2 get-ebs-default-kms-key-id
```

La sortie indique la clé CMK par défaut pour le chiffrement EBS, qui est une clé CMK AWS gérée avec un alias. `alias/aws/ebs`

```
{
    "KmsKeyId": "alias/aws/ebs"
}
```

La sortie suivante montre une clé CMK personnalisée pour le chiffrement EBS.

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous GetEbsDefaultKmsKeyIdà la section Référence des AWS CLI commandes.

`get-ebs-encryption-by-default`

L'exemple de code suivant montre comment utiliser`get-ebs-encryption-by-default`.

AWS CLI

Pour indiquer si le chiffrement EBS est activé par défaut

L'`get-ebs-encryption-by-default`exemple suivant indique si le chiffrement EBS est activé par défaut pour votre AWS compte dans la région actuelle.

```
aws ec2 get-ebs-encryption-by-default
```

Le résultat suivant indique que le chiffrement EBS est désactivé par défaut.

```
{
    "EbsEncryptionByDefault": false
}
```

Le résultat suivant indique que le chiffrement EBS est activé par défaut.

```
{
    "EbsEncryptionByDefault": true
}
```

- Pour plus de détails sur l'API, reportez-vous GetEbsEncryptionByDefaultà la section Référence des AWS CLI commandes.

`get-flow-logs-integration-template`

L'exemple de code suivant montre comment utiliser`get-flow-logs-integration-template`.

AWS CLI

Pour créer un CloudFormation modèle afin d'automatiser l'intégration des journaux de flux VPC à Amazon Athena

Les `get-flow-logs-integration-template` exemples suivants créent un CloudFormation modèle pour automatiser l'intégration des journaux de flux VPC à Amazon Athena.

Linux :

```
aws ec2 get-flow-logs-integration-template \
    --flow-log-id fl-1234567890abcdef0 \
    --config-delivery-s3-destination-arn arn:aws:s3:::amzn-s3-demo-bucket \
    --integrate-services
 AthenaIntegrations='[{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

Windows :

```
aws ec2 get-flow-logs-integration-template ^
    --flow-log-id fl-1234567890abcdef0 ^
    --config-delivery-s3-destination-arn arn:aws:s3:::amzn-s3-demo-bucket ^
    --integrate-
services AthenaIntegrations=[{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-
s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

Sortie :

```
{
```

```
    "Result": "https://amzn-s3-demo-bucket.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_fl-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
}
```

Pour plus d'informations sur l'utilisation CloudFormation des modèles, voir Utilisation des AWS CloudFormation modèles dans le Guide de AWS CloudFormation l'utilisateur.

Pour plus d'informations sur l'utilisation d'Amazon Athena et des journaux de flux, consultez la section Query flow logs using Amazon Athena dans le manuel Amazon Virtual Private Cloud User Guide.

- Pour plus de détails sur l'API, reportez-vous GetFlowLogsIntegrationTemplateà la section Référence des AWS CLI commandes.

`get-groups-for-capacity-reservation`

L'exemple de code suivant montre comment utiliser`get-groups-for-capacity-reservation`.

AWS CLI

Pour répertorier les groupes de ressources faisant l'objet d'une réservation de capacité

L'`get-groups-for-capacity-reservation`exemple suivant répertorie les groupes de ressources auxquels la réservation de capacité spécifiée a été ajoutée.

```
aws ec2 get-groups-for-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Sortie :

```
{
    "CapacityReservationsGroup": [
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
            "OwnerId": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes de réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetGroupsForCapacityReservationà la section Référence des AWS CLI commandes.

`get-host-reservation-purchase-preview`

L'exemple de code suivant montre comment utiliser`get-host-reservation-purchase-preview`.

AWS CLI

Pour obtenir un aperçu de l'achat d'une réservation d'hôte dédié

Cet exemple fournit un aperçu des coûts d'une réservation d'hôte dédié spécifiée pour l'hôte dédié spécifié dans votre compte.

Commande :

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

Sortie :

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
      {
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- Pour plus de détails sur l'API, reportez-vous GetHostReservationPurchasePreviewà la section Référence des AWS CLI commandes.

`get-image-block-public-access-state`

L'exemple de code suivant montre comment utiliser`get-image-block-public-access-state`.

AWS CLI

Pour obtenir l'état de blocage de l'accès public pour AMIs la région spécifiée

L'`get-image-block-public-access-state`exemple suivant permet d'obtenir l'état de blocage de l'accès public AMIs au niveau du compte dans la région spécifiée.

```
aws ec2 get-image-block-public-access-state \
    --region us-east-1
```

Sortie :

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

Pour plus d'informations, consultez Bloquer l'accès public à votre compte AMIs dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetImageBlockPublicAccessStateà la section Référence des AWS CLI commandes.

`get-instance-types-from-instance-requirements`

L'exemple de code suivant montre comment utiliser`get-instance-types-from-instance-requirements`.

AWS CLI

Pour prévisualiser les types d'instances correspondant aux attributs spécifiés

L'`get-instance-types-from-instance-requirements`exemple suivant génère d'abord une liste de tous les attributs possibles pouvant être spécifiés à l'aide du `--generate-cli-skeleton`on paramètre, puis enregistre la liste dans un fichier JSON. Le fichier JSON est

ensuite utilisé pour personnaliser les attributs pour lesquels prévisualiser les types d'instances correspondants.

Pour générer tous les attributs possibles et enregistrer la sortie directement dans un fichier JSON, utilisez la commande suivante.

```
aws ec2 get-instance-types-from-instance-requirements \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

Sortie :

```
{
    "DryRun": true,
    "ArchitectureTypes": [
        "x86_64_mac"
    ],
    "VirtualizationTypes": [
        "paravirtual"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 0,
            "Max": 0
        },
        "MemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "CpuManufacturers": [
            "intel"
        ],
        "MemoryGiBPerVCpu": {
            "Min": 0.0,
            "Max": 0.0
        },
        "ExcludedInstanceTypes": [
            ""
        ],
        "InstanceGenerations": [
            "current"
        ],
        "SpotMaxPricePercentageOverLowestPrice": 0,
```

```
        "OnDemandMaxPricePercentageOverLowestPrice": 0,
        "BareMetal": "included",
        "BurstablePerformance": "excluded",
        "RequireHibernateSupport": true,
        "NetworkInterfaceCount": {
            "Min": 0,
            "Max": 0
        },
        "LocalStorage": "required",
        "LocalStorageTypes": [
            "hdd"
        ],
        "TotalLocalStorageGB": {
            "Min": 0.0,
            "Max": 0.0
        },
        "BaselineEbsBandwidthMbps": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorTypes": [
            "inference"
        ],
        "AcceleratorCount": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorManufacturers": [
            "xilinx"
        ],
        "AcceleratorNames": [
            "t4"
        ],
        "AcceleratorTotalMemoryMiB": {
            "Min": 0,
            "Max": 0
        }
    },
    "MaxResults": 0,
    "NextToken": ""
}
```

Configurez le fichier JSON. Vous devez fournir des valeurs pour `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` et `MemoryMiB`. Vous pouvez omettre les autres attributs. En cas d'omission, les valeurs par défaut sont utilisées. Pour une description de chaque attribut et de leurs valeurs par défaut, consultez get-instance-types-from -instance-requirements < - - requirements.html> https://docs.aws.amazon.com/cli/latest/reference/ec2/get. instance-types-from-instance

Prévisualisez les types d'instances dont les attributs sont spécifiés dans`attributes.json`. Spécifiez le nom et le chemin d'accès à votre fichier JSON à l'aide du `--cli-input-json` paramètre. Dans la demande suivante, la sortie est formatée sous forme de tableau.

```
aws ec2 get-instance-types-from-instance-requirements \
    --cli-input-json file://attributes.json \
    --output table
```

Contenu du `attributes.json` fichier :

```
{

    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

Sortie :

```
-----------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+---------------------------------------+
||             InstanceTypes             ||
|+-------------------------------------+|
||             InstanceType              ||
|+-------------------------------------+|
||  c4.xlarge                          ||
||  c5.xlarge                          ||
||  c5a.xlarge                         ||
||  c5ad.xlarge                        ||
||  c5d.xlarge                         ||
||  c5n.xlarge                         ||
||  d2.xlarge                          ||
...
```

Pour plus d'informations sur la sélection du type d'instance basée sur les attributs, consultez Comment fonctionne la sélection du type d'instance basée sur les attributs dans le guide de l'utilisateur Amazon. EC2

- Pour plus de détails sur l'API, reportez-vous GetInstanceTypesFromInstanceRequirementsà la section Référence des AWS CLI commandes.

`get-instance-uefi-data`

L'exemple de code suivant montre comment utiliser`get-instance-uefi-data`.

AWS CLI

Pour récupérer des données UEFI à partir d'une instance

L'`get-instance-uefi-data`exemple suivant récupère les données UEFI d'une instance. Si la sortie est vide, l'instance ne contient pas de données UEFI.

```
aws ec2 get-instance-uefi-data \
    --instance-id i-0123456789example
```

Sortie :

```
{
    "InstanceId": "i-0123456789example",
```

```
    "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.
    <snipped>
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/AODshho="
}
```

Pour plus d'informations, consultez UEFI Secure Boot dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetInstanceUefiDataà la section Référence des AWS CLI commandes.

`get-ipam-address-history`

L'exemple de code suivant montre comment utiliser`get-ipam-address-history`.

AWS CLI

Pour obtenir l'historique d'un CIDR

L'`get-ipam-address-history`exemple suivant permet d'obtenir l'historique d'un CIDR.

(Linux) :

```
aws ec2 get-ipam-address-history \
   --cidr 10.0.0.0/16 \
   --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
   --start-time 2021-12-08T01:00:00.000Z \
   --end-time 2021-12-10T01:00:00.000Z
```

(Fenêtres) :

```
aws ec2 get-ipam-address-history ^
   --cidr 10.0.0.0/16 ^
   --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
   --start-time 2021-12-08T01:00:00.000Z ^
   --end-time 2021-12-10T01:00:00.000Z
```

Sortie :

```
{
    "HistoryRecords": [
```

```
        {
            "ResourceOwnerId": "123456789012",
            "ResourceRegion": "us-west-1",
            "ResourceType": "vpc",
            "ResourceId": "vpc-06cbefa9ee907e1c0",
            "ResourceCidr": "10.0.0.0/16",
            "ResourceName": "Demo",
            "ResourceComplianceStatus": "unmanaged",
            "ResourceOverlapStatus": "overlapping",
            "VpcId": "vpc-06cbefa9ee907e1c0",
            "SampledStartTime": "2021-12-08T19:54:57.675000+00:00"
        },
        {
            "ResourceOwnerId": "123456789012",
            "ResourceRegion": "us-east-2",
            "ResourceType": "vpc",
            "ResourceId": "vpc-042702f474812c9ad",
            "ResourceCidr": "10.0.0.0/16",
            "ResourceName": "test",
            "ResourceComplianceStatus": "unmanaged",
            "ResourceOverlapStatus": "overlapping",
            "VpcId": "vpc-042702f474812c9ad",
            "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"
        },
        {
            "ResourceOwnerId": "123456789012",
            "ResourceRegion": "us-east-2",
            "ResourceType": "vpc",
            "ResourceId": "vpc-042b8a44f64267d67",
            "ResourceCidr": "10.0.0.0/16",
            "ResourceName": "tester",
            "ResourceComplianceStatus": "unmanaged",
            "ResourceOverlapStatus": "overlapping",
            "VpcId": "vpc-042b8a44f64267d67",
            "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"
        }
    ]
}
```


Pour plus d'informations, consultez la section Afficher l'historique des adresses IP dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous GetIpamAddressHistoryà la section Référence des AWS CLI commandes.

`get-ipam-discovered-accounts`

L'exemple de code suivant montre comment utiliser`get-ipam-discovered-accounts`.

AWS CLI

Pour consulter les comptes découverts par un IPAM

Dans ce scénario, vous êtes un administrateur délégué IPAM qui souhaite consulter les AWS comptes propriétaires des ressources découvertes par l'IPAM.

`--discovery-region`Il s'agit de la région d'exploitation IPAM dans laquelle vous souhaitez consulter les statuts des comptes surveillés. Par exemple, si vous avez trois régions d'exploitation IPAM, vous souhaiterez peut-être effectuer cette demande trois fois pour afficher les horodatages spécifiques à la découverte dans chacune de ces régions en particulier.

L'`get-ipam-discovered-accounts`exemple suivant répertorie les AWS comptes qui possèdent les ressources découvertes par l'IPAM.

```
aws ec2 get-ipam-discovered-accounts \
   --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
   --discovery-region us-east-1
```

Sortie :

```
{
    "IpamDiscoveredAccounts": [
        {
            "AccountId": "149977607591",
            "DiscoveryRegion": "us-east-1",
            "LastAttemptedDiscoveryTime": "2024-02-09T19:04:31.379000+00:00",
            "LastSuccessfulDiscoveryTime": "2024-02-09T19:04:31.379000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez Intégrer l'IPAM à des comptes extérieurs à votre organisation dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous GetIpamDiscoveredAccountsà la section Référence des AWS CLI commandes.

`get-ipam-discovered-public-addresses`

L'exemple de code suivant montre comment utiliser`get-ipam-discovered-public-addresses`.

AWS CLI

Pour afficher les adresses IP publiques découvertes

Dans cet exemple, vous êtes un administrateur délégué IPAM et vous souhaitez consulter les adresses IP des ressources découvertes par l'IPAM. Vous pouvez obtenir l'ID de découverte des ressources avec describe-ipam-resource-discoveries.

L'`get-ipam-discovered-public-addresses`exemple suivant montre les adresses IP publiques découvertes pour la découverte d'une ressource.

```
aws ec2 get-ipam-discovered-public-addresses \
    --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
    --address-region us-east-1 \
    --region us-east-1
```

Sortie :

```
{
    "IpamDiscoveredPublicAddresses": [
        {
           "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
            "AddressRegion": "us-east-1",
            "Address": "54.208.155.7",
            "AddressOwnerId": "320805250157",
            "AssociationStatus": "associated",
            "AddressType": "ec2-public-ip",
            "VpcId": "vpc-073b294916198ce49",
            "SubnetId": "subnet-0b6c8a8839e9a4f15",
            "NetworkInterfaceId": "eni-081c446b5284a5e06",
            "NetworkInterfaceDescription": "",
            "InstanceId": "i-07459a6fca5b35823",
            "Tags": {},
            "NetworkBorderGroup": "us-east-1c",
            "SecurityGroups": [
                {
                    "GroupName": "launch-wizard-2",
                    "GroupId": "sg-0a489dd6a65c244ce"
                }
```

```
            ],
            "SampleTime": "2024-04-05T15:13:59.228000+00:00"
        },
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
            "AddressRegion": "us-east-1",
            "Address": "44.201.251.218",
            "AddressOwnerId": "470889052923",
            "AssociationStatus": "associated",
            "AddressType": "ec2-public-ip",
            "VpcId": "vpc-6c31a611",
            "SubnetId": "subnet-062f47608b99834b1",
            "NetworkInterfaceId": "eni-024845359c2c3ae9b",
            "NetworkInterfaceDescription": "",
            "InstanceId": "i-04ef786d9c4e03f41",
            "Tags": {},
            "NetworkBorderGroup": "us-east-1a",
            "SecurityGroups": [
                {
                    "GroupName": "launch-wizard-32",
                    "GroupId": "sg-0ed1a426e96a68374"
                }
            ],
            "SampleTime": "2024-04-05T15:13:59.145000+00:00"
        }
}
```


Pour plus d'informations, consultez la section Afficher les informations IP publiques dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous GetIpamDiscoveredPublicAddressesà la section Référence des AWS CLI commandes.

`get-ipam-discovered-resource-cidrs`

L'exemple de code suivant montre comment utiliser`get-ipam-discovered-resource-cidrs`.

AWS CLI

Pour afficher l'adresse IP CIDRs découverte par un IPAM

Dans cet exemple, vous êtes un administrateur délégué IPAM qui souhaite consulter les détails relatifs à l'adresse IP CIDRs des ressources découvertes par l'IPAM.

Pour compléter cette demande :

La découverte de ressources que vous choisissez doit être associée à l'IPAM. Il `--resource-region` s'agit de la AWS région dans laquelle la ressource a été créée.

L'`get-ipam-discovered-resource-cidrs`exemple suivant répertorie les adresses IP des ressources découvertes par l'IPAM.

```
aws ec2 get-ipam-discovered-resource-cidrs \
   --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
   --resource-region us-east-1
```

Sortie :

```
{
    {
        "IpamDiscoveredResourceCidrs": [
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "ResourceRegion": "us-east-1",
            "ResourceId": "vpc-0c974c95ca7ceef4a",
            "ResourceOwnerId": "149977607591",
            "ResourceCidr": "172.31.0.0/16",
            "ResourceType": "vpc",
            "ResourceTags": [],
            "IpUsage": 0.375,
            "VpcId": "vpc-0c974c95ca7ceef4a",
            "SampleTime": "2024-02-09T19:15:16.529000+00:00"
        },
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "ResourceRegion": "us-east-1",
            "ResourceId": "subnet-07fe028119082a8c1",
            "ResourceOwnerId": "149977607591",
            "ResourceCidr": "172.31.0.0/20",
            "ResourceType": "subnet",
            "ResourceTags": [],
            "IpUsage": 0.0012,
            "VpcId": "vpc-0c974c95ca7ceef4a",
            "SampleTime": "2024-02-09T19:15:16.529000+00:00"
        },
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
```

```
            "ResourceRegion": "us-east-1",
            "ResourceId": "subnet-0a96893763984cc4e",
            "ResourceOwnerId": "149977607591",
            "ResourceCidr": "172.31.64.0/20",
            "ResourceType": "subnet",
            "ResourceTags": [],
            "IpUsage": 0.0012,
            "VpcId": "vpc-0c974c95ca7ceef4a",
            "SampleTime": "2024-02-09T19:15:16.529000+00:00"
        }
    }
}
```

Pour plus d'informations, consultez la section Surveiller l'utilisation du CIDR par ressource dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous GetIpamDiscoveredResourceCidrsà la section Référence des AWS CLI commandes.

`get-ipam-pool-allocations`

L'exemple de code suivant montre comment utiliser`get-ipam-pool-allocations`.

AWS CLI

Pour obtenir le CIDRs montant alloué à partir d'un pool IPAM

L'`get-ipam-pool-allocations`exemple suivant obtient le CIDRs montant alloué à partir d'un pool IPAM.

(Linux) :

```
aws ec2 get-ipam-pool-allocations \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --filters Name=ipam-pool-allocation-id,Values=ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220
```

(Fenêtres) :

```
aws ec2 get-ipam-pool-allocations ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
```

```
    --filters Name=ipam-pool-allocation-id,Values=ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220
```

Sortie :

```
{
    "IpamPoolAllocations": [
        {
            "Cidr": "10.0.0.0/16",
            "IpamPoolAllocationId": "ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220",
            "ResourceType": "custom",
            "ResourceOwner": "123456789012"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetIpamPoolAllocationsà la section Référence des AWS CLI commandes.

`get-ipam-pool-cidrs`

L'exemple de code suivant montre comment utiliser`get-ipam-pool-cidrs`.

AWS CLI

Pour transférer le CIDRs provisionné vers un pool IPAM

Dans l'`get-ipam-pool-cidrs`exemple suivant, le CIDRs provisionnement est transféré à un pool IPAM.

(Linux) :

```
aws ec2 get-ipam-pool-cidrs \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --filters 'Name=cidr,Values=10.*'
```

(Fenêtres) :

```
aws ec2 get-ipam-pool-cidrs ^
```

```
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --filters Name=cidr,Values=10.*
```

Sortie :

```
{
    "IpamPoolCidr": {
        "Cidr": "10.0.0.0/24",
        "State": "provisioned"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetIpamPoolCidrsà la section Référence des AWS CLI commandes.

`get-ipam-resource-cidrs`

L'exemple de code suivant montre comment utiliser`get-ipam-resource-cidrs`.

AWS CLI

Pour obtenir le CIDRs montant alloué à une ressource

L'`get-ipam-resource-cidrs`exemple suivant obtient le CIDRs montant alloué à une ressource.

(Linux) :

```
aws ec2 get-ipam-resource-cidrs \
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
    --filters Name=management-state,Values=unmanaged
```

(Fenêtres) :

```
aws ec2 get-ipam-resource-cidrs ^
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
    --filters Name=management-state,Values=unmanaged
```

Sortie :

```
{
    "IpamResourceCidrs": [
        {
            "IpamId": "ipam-08440e7a3acde3908",
            "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",
            "ResourceRegion": "us-east-2",
            "ResourceOwnerId": "123456789012",
            "ResourceId": "vpc-621b8709",
            "ResourceName": "Default AWS VPC",
            "ResourceCidr": "172.33.0.0/16",
            "ResourceType": "vpc",
            "ResourceTags": [
                {
                    "Key": "Environment",
                    "Value": "Test"
                },
                {
                    "Key": "Name",
                    "Value": "Default AWS VPC"
                }
            ],
            "IpUsage": 0.0039,
            "ComplianceStatus": "unmanaged",
            "ManagementState": "unmanaged",
            "OverlapStatus": "nonoverlapping",
            "VpcId": "vpc-621b8709"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveiller l'utilisation du CIDR par ressource dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous GetIpamResourceCidrsà la section Référence des AWS CLI commandes.

`get-launch-template-data`

L'exemple de code suivant montre comment utiliser`get-launch-template-data`.

AWS CLI

Pour obtenir des données d'instance pour un modèle de lancement

Cet exemple obtient des données sur l'instance spécifiée et utilise l'`--query`option permettant de renvoyer le contenu`LaunchTemplateData`. Vous pouvez utiliser la sortie comme base pour créer un modèle de lancement ou une version de modèle de lancement.

Commande :

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
 'LaunchTemplateData'
```

Sortie :

```
{
      "Monitoring": {},
      "ImageId": "ami-8c1be5f6",
      "BlockDeviceMappings": [
          {
              "DeviceName": "/dev/xvda",
              "Ebs": {
                  "DeleteOnTermination": true
              }
          }
      ],
      "EbsOptimized": false,
      "Placement": {
          "Tenancy": "default",
          "GroupName": "",
          "AvailabilityZone": "us-east-1a"
      },
      "InstanceType": "t2.micro",
      "NetworkInterfaces": [
          {
              "Description": "",
              "NetworkInterfaceId": "eni-35306abc",
              "PrivateIpAddresses": [
                  {
                      "Primary": true,
                      "PrivateIpAddress": "10.0.0.72"
                  }
              ],
              "SubnetId": "subnet-7b16de0c",
              "Groups": [
                  "sg-7c227019"
              ],
```

```
            "Ipv6Addresses": [
                {
                    "Ipv6Address": "2001:db8:1234:1a00::123"
                }
            ],
            "PrivateIpAddress": "10.0.0.72"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLaunchTemplateDataà la section Référence des AWS CLI commandes.

`get-managed-prefix-list-associations`

L'exemple de code suivant montre comment utiliser`get-managed-prefix-list-associations`.

AWS CLI

Pour obtenir des associations de listes de préfixes

L'`get-managed-prefix-list-associations`exemple suivant obtient les ressources associées à la liste de préfixes spécifiée.

```
aws ec2 get-managed-prefix-list-associations \
    --prefix-list-id pl-0123456abcabcabc1
```

Sortie :

```
{
    "PrefixListAssociations": [
        {
            "ResourceId": "sg-0abc123456abc12345",
            "ResourceOwner": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous GetManagedPrefixListAssociationsà la section Référence des AWS CLI commandes.

`get-managed-prefix-list-entries`

L'exemple de code suivant montre comment utiliser`get-managed-prefix-list-entries`.

AWS CLI

Pour obtenir les entrées d'une liste de préfixes

Ce qui suit `get-managed-prefix-list-entries` permet d'obtenir les entrées de la liste de préfixes spécifiée.

```
aws ec2 get-managed-prefix-list-entries \
    --prefix-list-id pl-0123456abcabcabc1
```

Sortie :

```
{
    "Entries": [
        {
            "Cidr": "10.0.0.0/16",
            "Description": "vpc-a"
        },
        {
            "Cidr": "10.2.0.0/16",
            "Description": "vpc-b"
        }
    ]
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous GetManagedPrefixListEntriesà la section Référence des AWS CLI commandes.

`get-network-insights-access-scope-analysis-findings`

L'exemple de code suivant montre comment utiliser`get-network-insights-access-scope-analysis-findings`.

AWS CLI

Pour obtenir les résultats de l'analyse du périmètre d'accès de Network Insights

L'`get-network-insights-access-scope-analysis-findings`exemple suivant permet d'obtenir les résultats de l'analyse de portée sélectionnée dans votre AWS compte.

```
aws ec2 get-network-insights-access-scope-analysis-findings \
    --region us-east-1 \
    --network-insights-access-scope-analysis-id nis \
    --nis-123456789111
```

Sortie :

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
    "AnalysisFindings": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
            "NetworkInsightsAccessScopeId": "nis-123456789111",
            "FindingComponents": [
                {
                    "SequenceNumber": 1,
                    "Component": {
                        "Id": "eni-02e3d42d5cceca67d",
                        "Arn": "arn:aws:ec2:us-east-1:936459623503:network-
interface/eni-02e3d32d9cceca17d"
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": [
                            "0.0.0.0/5",
                            "11.0.0.0/8",
                            "12.0.0.0/6",
                            "128.0.0.0/3",
                            "16.0.0.0/4",
                            "160.0.0.0/5",
                            "168.0.0.0/6",
                            "172.0.0.0/12"
```

```
                        "8.0.0.0/7"
                    ],
                    "DestinationPortRanges": [
                        {
                            "From": 0,
                            "To": 65535
                        }
                    ],
                    "Protocol": "6",
                    "SourceAddresses": [
                        "10.0.2.253/32"
                    ],
                    "SourcePortRanges": [
                        {
                            "From": 0,
                            "To": 65535
                        }
                    ]
                }, [etc]
            ]
        }
    }
  ]
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous GetNetworkInsightsAccessScopeAnalysisFindingsà la section Référence des AWS CLI commandes.

`get-network-insights-access-scope-content`

L'exemple de code suivant montre comment utiliser`get-network-insights-access-scope-content`.

AWS CLI

Pour obtenir le contenu du périmètre d'accès à Network Insights

L'`get-network-insights-access-scope-content`exemple suivant permet d'obtenir le contenu de l'ID d'analyse de portée sélectionné dans votre AWS compte.

```
aws ec2 get-network-insights-access-scope-content \
    --region us-east-1 \
    --network-insights-access-scope-id nis-123456789222
```

Sortie :

```
{
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-123456789222",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::NetworkInterface"
                        ]
                    }
                },
                "Destination": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous GetNetworkInsightsAccessScopeContentà la section Référence des AWS CLI commandes.

`get-password-data`

L'exemple de code suivant montre comment utiliser`get-password-data`.

AWS CLI

Pour obtenir le mot de passe crypté

Cet exemple permet d'obtenir le mot de passe crypté.

Commande :

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

Sortie :

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-08-07T22:18:38.000Z",
    "PasswordData": "gSlJFq+VpcZXqy+iktxMF6NyxQ4qCrT4+gaOuNOenX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTYP7WmU3TUnhsuBd+p6LVk7T2lKUml6OXbk6WPW1VYYm/TRPB1
e1DQ7PY4an/DgZT4mwcpRFigzhniQgDDeO1InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVelVz/
53TkDtxbNoU606M1gK9zUWSxqEgwvbV2j8c5rP0WCuaMWSFl4ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yLlf3etP2L4ZR6CvG7K1hx7VKOQVN32Dajw=="
}
```

Pour obtenir le mot de passe déchiffré

Cet exemple permet d'obtenir le mot de passe déchiffré.

Commande :

```
aws ec2 get-password-data --instance-id  i-1234567890abcdef0 --priv-launch-key C:
\Keys\MyKeyPair.pem
```

Sortie :

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-08-30T23:18:05.000Z",
    "PasswordData": "&ViJ652e*u"
}
```

- Pour plus de détails sur l'API, reportez-vous GetPasswordDataà la section Référence des AWS CLI commandes.

`get-reserved-instances-exchange-quote`

L'exemple de code suivant montre comment utiliser`get-reserved-instances-exchange-quote`.

AWS CLI

Pour obtenir un devis pour l'échange d'une instance réservée convertible

Cet exemple permet d'obtenir les informations d'échange pour les instances réservées convertibles spécifiées.

Commande :

```
aws ec2 get-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

Sortie :

```
{
  "CurrencyCode": "USD",
  "ReservedInstanceValueSet": [
      {
          "ReservedInstanceId": "7b8750c3-397e-4da4-bbcb-a45ebexample",
          "ReservationValue": {
              "RemainingUpfrontValue": "0.000000",
              "HourlyPrice": "0.027800",
              "RemainingTotalValue": "730.556200"
          }
      }
  ],
  "PaymentDue": "424.983828",
  "TargetConfigurationValueSet": [
      {
          "TargetConfiguration": {
              "InstanceCount": 5,
              "OfferingId": "6fea5434-b379-434c-b07b-a7abexample"
          },
          "ReservationValue": {
              "RemainingUpfrontValue": "424.983828",
              "HourlyPrice": "0.016000",
              "RemainingTotalValue": "845.447828"
```

```
            }
        }
    ],
    "IsValidExchange": true,
    "OutputReservedInstancesWillExpireAt": "2020-10-01T13:03:39Z",
    "ReservedInstanceValueRollup": {
        "RemainingUpfrontValue": "0.000000",
        "HourlyPrice": "0.027800",
        "RemainingTotalValue": "730.556200"
    },
    "TargetConfigurationValueRollup": {
        "RemainingUpfrontValue": "424.983828",
        "HourlyPrice": "0.016000",
        "RemainingTotalValue": "845.447828"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetReservedInstancesExchangeQuoteà la section Référence des AWS CLI commandes.

`get-security-groups-for-vpc`

L'exemple de code suivant montre comment utiliser`get-security-groups-for-vpc`.

AWS CLI

Pour afficher les groupes de sécurité qui peuvent être associés aux interfaces réseau dans un VPC spécifique.

L'`get-security-groups-for-vpc`exemple suivant montre les groupes de sécurité qui peuvent être associés aux interfaces réseau dans le VPC.

```
aws ec2 get-security-groups-for-vpc \
    --vpc-id vpc-6c31a611 \
    --region us-east-1
```

Sortie :

```
{
    "SecurityGroupForVpcs": [
        {
            "Description": "launch-wizard-36 created 2022-08-29T15:59:35.338Z",
```

```
            "GroupName": "launch-wizard-36",
            "OwnerId": "470889052923",
            "GroupId": "sg-007e0c3027ee885f5",
            "Tags": [],
            "PrimaryVpcId": "vpc-6c31a611"
        },
        {
            "Description": "launch-wizard-18 created 2024-01-19T20:22:27.527Z",
            "GroupName": "launch-wizard-18",
            "OwnerId": "470889052923",
            "GroupId": "sg-0147193bef51c9eef",
            "Tags": [],
            "PrimaryVpcId": "vpc-6c31a611"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSecurityGroupsForVpcà la section Référence des AWS CLI commandes.

`get-serial-console-access-status`

L'exemple de code suivant montre comment utiliser`get-serial-console-access-status`.

AWS CLI

Pour consulter l'état de l'accès du compte à la console série

L'`get-serial-console-access-status`exemple suivant détermine si l'accès à la console série est activé pour votre compte.

```
aws ec2 get-serial-console-access-status
```

Sortie :

```
{
    "SerialConsoleAccessEnabled": true
}
```

Pour plus d'informations, consultez EC2 Serial Console dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetSerialConsoleAccessStatusà la section Référence des AWS CLI commandes.

`get-snapshot-block-public-access-state`

L'exemple de code suivant montre comment utiliser`get-snapshot-block-public-access-state`.

AWS CLI

Pour connaître l'état actuel du blocage de l'accès public aux instantanés

L'`get-snapshot-block-public-access-state`exemple suivant montre l'état actuel du blocage de l'accès public pour les instantanés.

```
aws ec2 get-snapshot-block-public-access-state
```

Sortie :

```
{
    "State": "block-all-sharing"
}
```

Pour plus d'informations, consultez Bloquer l'accès public aux instantanés dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous GetSnapshotBlockPublicAccessStateà la section Référence des AWS CLI commandes.

`get-spot-placement-scores`

L'exemple de code suivant montre comment utiliser`get-spot-placement-scores`.

AWS CLI

Pour calculer le score de placement Spot pour des exigences spécifiques

L'`get-spot-placement-scores`exemple suivant génère d'abord une liste de tous les paramètres possibles qui peuvent être spécifiés pour la configuration du score de placement Spot à l'aide du `--generate-cli-skeleton`paramètre, puis enregistre la liste dans un fichier JSON. Le fichier JSON est ensuite utilisé pour configurer les exigences à utiliser pour calculer le score de placement Spot.

Pour générer tous les paramètres possibles pouvant être spécifiés pour la configuration du score de placement Spot, et enregistrer la sortie directement dans un fichier JSON.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

Sortie :

```
{
    "InstanceTypes": [
        ""
    ],
    "TargetCapacity": 0,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "RegionNames": [
        ""
    ],
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": [
            "x86_64_mac"
        ],
        "VirtualizationTypes": [
            "hvm"
        ],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 0,
                "Max": 0
            },
            "MemoryMiB": {
                "Min": 0,
                "Max": 0
            },
            "CpuManufacturers": [
                "amd"
            ],
            "MemoryGiBPerVCpu": {
                "Min": 0.0,
                "Max": 0.0
            },
            "ExcludedInstanceTypes": [
                ""
            ],
            "InstanceGenerations": [
```

```
                "previous"
            ],
            "SpotMaxPricePercentageOverLowestPrice": 0,
            "OnDemandMaxPricePercentageOverLowestPrice": 0,
            "BareMetal": "excluded",
            "BurstablePerformance": "excluded",
            "RequireHibernateSupport": true,
            "NetworkInterfaceCount": {
                "Min": 0,
                "Max": 0
            },
            "LocalStorage": "included",
            "LocalStorageTypes": [
                "hdd"
            ],
            "TotalLocalStorageGB": {
                "Min": 0.0,
                "Max": 0.0
            },
            "BaselineEbsBandwidthMbps": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorTypes": [
                "fpga"
            ],
            "AcceleratorCount": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorManufacturers": [
                "amd"
            ],
            "AcceleratorNames": [
                "vu9p"
            ],
            "AcceleratorTotalMemoryMiB": {
                "Min": 0,
                "Max": 0
            }
        }
    },
    "DryRun": true,
    "MaxResults": 0,
```

```
    "NextToken": ""
}
```

Configurez le fichier JSON. Vous devez fournir une valeur pour `TargetCapacity`. Pour une description de chaque paramètre et de leurs valeurs par défaut, voir Calculer le score de placement Spot (AWS CLI) <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/spotcalculate-sps-cli-placement-score.html# >.

Calculez le score de placement Spot pour les exigences spécifiées dans`attributes.json`. Spécifiez le nom et le chemin d'accès à votre fichier JSON à l'aide du `--cli-input-json` paramètre.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --cli-input-json file://attributes.json
```

Afficher si la valeur `SingleAvailabilityZone` est définie `false` ou omise (si elle est omise, la valeur par défaut est définie sur). `false` Une liste de régions notées est renvoyée.

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

Afficher si la `SingleAvailabilityZone` valeur est définie sur`true`. Une liste de SingleAvailability zones notées est renvoyée.

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "use1-az1"
        "Score": 8
    },
    {
```

```
        "Region": "us-east-1",
        "AvailabilityZoneId": "usw2-az3"
        "Score": 6
    },
   ...
```

Pour plus d'informations sur le calcul d'un score de placement Spot, et par exemple sur les configurations, consultez Calculer un score de placement Spot dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetSpotPlacementScoresà la section Référence des AWS CLI commandes.

`get-subnet-cidr-reservations`

L'exemple de code suivant montre comment utiliser`get-subnet-cidr-reservations`.

AWS CLI

Pour obtenir des informations sur la réservation d'un sous-réseau CIDR

L'`get-subnet-cidr-reservations`exemple suivant affiche des informations sur la réservation CIDR du sous-réseau spécifiée.

```
aws ec2 get-subnet-cidr-reservations \
    --subnet-id subnet-03c51e2e6cEXAMPLE
```

Sortie :

```
{
    "SubnetIpv4CidrReservations": [
        {
            "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
            "SubnetId": "subnet-03c51e2e6cEXAMPLE",
            "Cidr": "10.1.0.16/28",
            "ReservationType": "prefix",
            "OwnerId": "123456789012"
        }
    ],
    "SubnetIpv6CidrReservations": []
}
```

Pour plus d'informations, consultez Subnet CIDR reservations dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous GetSubnetCidrReservationsà la section Référence des AWS CLI commandes.

`get-transit-gateway-attachment-propagations`

L'exemple de code suivant montre comment utiliser`get-transit-gateway-attachment-propagations`.

AWS CLI

Pour répertorier les tables de routage vers lesquelles l'attachement de ressource spécifié propage les itinéraires

L'`get-transit-gateway-attachment-propagations`exemple suivant répertorie la table de routage vers laquelle l'attachement de ressource spécifié propage les itinéraires.

```
aws ec2 get-transit-gateway-attachment-propagations \
    --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE
```

Sortie :

```
{
    "TransitGatewayAttachmentPropagations": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
            "State": "enabled"
        }
    ]
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous GetTransitGatewayAttachmentPropagationsà la section Référence des AWS CLI commandes.

`get-transit-gateway-multicast-domain-associations`

L'exemple de code suivant montre comment utiliser`get-transit-gateway-multicast-domain-associations`.

AWS CLI

Pour afficher les informations relatives aux associations de domaines de multidiffusion de la passerelle de transit

L'`get-transit-gateway-multicast-domain-associations`exemple suivant renvoie les associations pour le domaine de multidiffusion spécifié.

```
aws ec2 get-transit-gateway-multicast-domain-associations \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Sortie :

```
{
    "MulticastDomainAssociations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8EXAMPLE",
            "ResourceId": "vpc-01128d2c24EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-000de86e3bEXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-4EXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
```

```
            "Subnet": {
                "SubnetId": "subnet-5EXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-aEXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-fEXAMPLE",
                "State": "associated"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Domaines de multidiffusion dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous GetTransitGatewayMulticastDomainAssociationsà la section Référence des AWS CLI commandes.

`get-transit-gateway-prefix-list-references`

L'exemple de code suivant montre comment utiliser`get-transit-gateway-prefix-list-references`.

AWS CLI

Pour obtenir les références d'une liste de préfixes dans une table de routage d'une passerelle de transit

L'`get-transit-gateway-prefix-list-references`exemple suivant obtient les références de liste de préfixes pour la table de routage de passerelle de transit spécifiée et filtre en fonction de l'ID d'une liste de préfixes spécifique.

```
aws ec2 get-transit-gateway-prefix-list-references \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --filters Name=prefix-list-id,Values=pl-11111122222222333
```

Sortie :

```
{
    "TransitGatewayPrefixListReferences": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
            "PrefixListId": "pl-11111122222222333",
            "PrefixListOwnerId": "123456789012",
            "State": "available",
            "Blackhole": false,
            "TransitGatewayAttachment": {
                "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
                "ResourceType": "vpc",
                "ResourceId": "vpc-112233445566aabbc"
            }
        }
    ]
}
```

Pour plus d'informations, voir les références aux listes de préfixes dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous GetTransitGatewayPrefixListReferencesà la section Référence des AWS CLI commandes.

`get-transit-gateway-route-table-associations`

L'exemple de code suivant montre comment utiliser`get-transit-gateway-route-table-associations`.

AWS CLI

Pour obtenir des informations sur les associations associées à la table de routage de la passerelle de transit spécifiée

L'`get-transit-gateway-route-table-associations`exemple suivant affiche des informations sur les associations pour la table de routage de passerelle de transit spécifiée.

```
aws ec2 get-transit-gateway-route-table-associations \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

Sortie :

```
{
    "Associations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
            "ResourceId": "vpc-4d7de228",
            "ResourceType": "vpc",
            "State": "associating"
        }
    ]
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous GetTransitGatewayRouteTableAssociationsà la section Référence des AWS CLI commandes.

`get-transit-gateway-route-table-propagations`

L'exemple de code suivant montre comment utiliser`get-transit-gateway-route-table-propagations`.

AWS CLI

Pour afficher des informations sur les propagations de la table de routage pour la table de routage de passerelle de transit spécifiée

L'`get-transit-gateway-route-table-propagations`exemple suivant renvoie les propagations de table de routage pour la table de routage spécifiée.

```
aws ec2 get-transit-gateway-route-table-propagations \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE
```

Sortie :

```
{
    "TransitGatewayRouteTablePropagations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
            "ResourceId": "vpc-3EXAMPLE",
            "ResourceType": "vpc",
            "State": "enabled"
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
            "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
            "ResourceType": "direct-connect-gateway",
            "State": "enabled"
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
            "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
            "ResourceType": "direct-connect-gateway",
            "State": "enabled"
        }
    ]
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous GetTransitGatewayRouteTablePropagationsà la section Référence des AWS CLI commandes.

`get-verified-access-endpoint-policy`

L'exemple de code suivant montre comment utiliser`get-verified-access-endpoint-policy`.

AWS CLI

Pour obtenir la politique d'accès vérifié d'un terminal

L'`get-verified-access-endpoint-policy`exemple suivant permet d'obtenir la politique d'accès vérifié du point de terminaison spécifié.

```
aws ec2 get-verified-access-endpoint-policy \
```

```
    --verified-access-endpoint-id vae-066fac616d4d546f2
```

Sortie :

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

Pour plus d'informations, consultez les politiques d'accès vérifié dans le guide de l'utilisateur de l'accès AWS vérifié.

- Pour plus de détails sur l'API, reportez-vous GetVerifiedAccessEndpointPolicyà la section Référence des AWS CLI commandes.

`get-verified-access-group-policy`

L'exemple de code suivant montre comment utiliser`get-verified-access-group-policy`.

AWS CLI

Pour obtenir la politique d'accès vérifié d'un groupe

L'`get-verified-access-group-policy`exemple suivant permet d'obtenir la politique d'accès vérifié du groupe spécifié.

```
aws ec2 get-verified-access-group-policy \
    --verified-access-group-id vagr-0dbe967baf14b7235
```

Sortie :

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

Pour plus d'informations, consultez la section Groupes d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous GetVerifiedAccessGroupPolicyà la section Référence des AWS CLI commandes.

`get-vpn-connection-device-sample-configuration`

L'exemple de code suivant montre comment utiliser`get-vpn-connection-device-sample-configuration`.

AWS CLI

Pour télécharger un exemple de fichier de configuration

L'`get-vpn-connection-device-sample-configuration`exemple suivant télécharge l'exemple de fichier de configuration spécifié. Pour répertorier les périphériques de passerelle avec un exemple de fichier de configuration, appelez la `get-vpn-connection-device-types` commande.

```
aws ec2 get-vpn-connection-device-sample-configuration \
    --vpn-connection-id vpn-123456789abc01234 \
    --vpn-connection-device-type-id 5fb390ba
```

Sortie :

```
{
    "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-file"
}
```

Pour plus d'informations, consultez Télécharger le fichier de configuration dans le guide de l'utilisateur du AWS Site-to-Site VPN.

- Pour plus de détails sur l'API, reportez-vous GetVpnConnectionDeviceSampleConfigurationà la section Référence des AWS CLI commandes.

`get-vpn-connection-device-types`

L'exemple de code suivant montre comment utiliser`get-vpn-connection-device-types`.

AWS CLI

Pour répertorier les périphériques de passerelle avec un exemple de fichier de configuration

L'`get-vpn-connection-device-types`exemple suivant répertorie les périphériques de passerelle de Palo Alto Networks dotés d'exemples de fichiers de configuration.

```
aws ec2 get-vpn-connection-device-types \
    --query "VpnConnectionDeviceTypes[?Vendor==`Palo Alto Networks`]"
```

Sortie :

```
[
    {
        "VpnConnectionDeviceTypeId": "754a6372",
        "Vendor": "Palo Alto Networks",
        "Platform": "PA Series",
        "Software": "PANOS 4.1.2+"
    },
    {
        "VpnConnectionDeviceTypeId": "9612cbed",
        "Vendor": "Palo Alto Networks",
        "Platform": "PA Series",
        "Software": "PANOS 4.1.2+ (GUI)"
    },
    {
        "VpnConnectionDeviceTypeId": "5fb390ba",
        "Vendor": "Palo Alto Networks",
        "Platform": "PA Series",
        "Software": "PANOS 7.0+"
    }
]
```

Pour plus d'informations, voir Télécharger le fichier de configuration dans le guide de l'utilisateur du AWS Site-to-Site VPN.

- Pour plus de détails sur l'API, reportez-vous GetVpnConnectionDeviceTypesà la section Référence des AWS CLI commandes.

`import-client-vpn-client-certificate-revocation-list`

L'exemple de code suivant montre comment utiliser`import-client-vpn-client-certificate-revocation-list`.

AWS CLI

Pour importer une liste de révocation de certificats clients

L'`import-client-vpn-client-certificate-revocation-list`exemple suivant importe une liste de révocation de certificats clients vers le point de terminaison VPN du client en spécifiant l'emplacement du fichier sur l'ordinateur local.

```
aws ec2 import-client-vpn-client-certificate-revocation-list \
    --certificate-revocation-list file:///path/to/crl.pem \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez les listes de révocation des certificats clients dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous ImportClientVpnClientCertificateRevocationListà la section Référence des AWS CLI commandes.

`import-image`

L'exemple de code suivant montre comment utiliser`import-image`.

AWS CLI

Pour importer un fichier image de machine virtuelle en tant qu'AMI

L'`import-image`exemple suivant importe le fichier OVA spécifié.

```
aws ec2 import-image \
```

```
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.ova}"
```

Sortie :

```
{
    "ImportTaskId": "import-ami-1234567890abcdef0",
    "Progress": "2",
    "SnapshotDetails": [
        {
            "DiskImageSize": 0.0,
            "Format": "ova",
            "UserBucket": {
                "S3Bucket": "my-import-bucket",
                "S3Key": "vms/my-server-vm.ova"
            }
        }
    ],
    "Status": "active",
    "StatusMessage": "pending"
}
```

- Pour plus de détails sur l'API, reportez-vous ImportImageà la section Référence des AWS CLI commandes.

`import-key-pair`

L'exemple de code suivant montre comment utiliser`import-key-pair`.

AWS CLI

Pour importer une clé publique

Commencez par générer une paire de clés avec l'outil de votre choix. Par exemple, utilisez cette commande ssh-keygen :

Commande :

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

Sortie :

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/my-key.
Your public key has been saved in /home/ec2-user/.ssh/my-key.pub.
...
```

Cet exemple de commande importe la clé publique spécifiée.

Commande :

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

Sortie :

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- Pour plus de détails sur l'API, reportez-vous ImportKeyPairà la section Référence des AWS CLI commandes.

`import-snapshot`

L'exemple de code suivant montre comment utiliser`import-snapshot`.

AWS CLI

Pour importer un instantané

L'`import-snapshot`exemple suivant importe le disque spécifié sous forme de capture instantanée.

```
aws ec2 import-snapshot \
    --description "My server VMDK" \
    --disk-container Format=VMDK,UserBucket={'S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.vmdk'}
```

Sortie :

```
{
    "Description": "My server VMDK",
    "ImportTaskId": "import-snap-1234567890abcdef0",
    "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "0.0",
        "Format": "VMDK",
        "Progress": "3",
        "Status": "active",
        "StatusMessage": "pending"
        "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.vmdk"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ImportSnapshotà la section Référence des AWS CLI commandes.

`list-images-in-recycle-bin`

L'exemple de code suivant montre comment utiliser`list-images-in-recycle-bin`.

AWS CLI

Pour répertorier les images dans la corbeille

L'`list-images-in-recycle-bin`exemple suivant répertorie toutes les images actuellement conservées dans la corbeille.

```
aws ec2 list-images-in-recycle-bin
```

Sortie :

```
{
    "Images": [
        {
            "RecycleBinEnterTime": "2022-03-14T15:35:08.000Z",
            "Description": "Monthly AMI One",
            "RecycleBinExitTime": "2022-03-15T15:35:08.000Z",
```

```
            "Name": "AMI_01",
            "ImageId": "ami-0111222333444abcd"
        }
    ]
}
```

Pour plus d'informations, consultez la section Récupérer les AMIs fichiers supprimés de la corbeille dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous ListImagesInRecycleBinà la section Référence des AWS CLI commandes.

`list-snapshots-in-recycle-bin`

L'exemple de code suivant montre comment utiliser`list-snapshots-in-recycle-bin`.

AWS CLI

Pour afficher des instantanés dans la corbeille

L'`list-snapshots-in-recycle-bin`exemple suivant répertorie les informations relatives aux instantanés contenus dans la corbeille, notamment l'ID de l'instantané, une description de l'instantané, l'ID du volume à partir duquel le cliché a été créé, la date et l'heure auxquelles le cliché a été supprimé et il est entré dans la corbeille, ainsi que la date et l'heure auxquelles la période de conservation expire.

```
aws ec2 list-snapshots-in-recycle-bin \
    --snapshot-id snap-01234567890abcdef
```

Sortie :

```
{
    "SnapshotRecycleBinInfo": [
        {
            "Description": "Monthly data backup snapshot",
            "RecycleBinEnterTime": "2022-12-01T13:00:00.000Z",
            "RecycleBinExitTime": "2022-12-15T13:00:00.000Z",
            "VolumeId": "vol-abcdef09876543210",
            "SnapshotId": "snap-01234567890abcdef"
        }
    ]
```

```
}
```

Pour plus d'informations sur la corbeille, consultez la section Récupérer des instantanés supprimés de la corbeille dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous ListSnapshotsInRecycleBinà la section Référence des AWS CLI commandes.

`lock-snapshot`

L'exemple de code suivant montre comment utiliser`lock-snapshot`.

AWS CLI

Exemple 1 : pour verrouiller un instantané en mode gouvernance

L'`lock-snapshot`exemple suivant verrouille le snapshot spécifié en mode gouvernance.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d \
    --lock-mode governance \
    --lock-duration 365
```

Sortie :

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d",
    "LockState": "governance",
    "LockDuration": 365,
    "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",
    "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00",
    "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00"
}
```

Pour plus d'informations, consultez Snapshot Lock dans le guide de l'utilisateur d'Amazon EBS.

Exemple 2 : pour verrouiller un instantané en mode de conformité

L'`lock-snapshot`exemple suivant verrouille le cliché spécifié en mode de conformité.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0163a8524c5b9901f \
```

```
    --lock-mode compliance \
    --cool-off-period 24 \
    --lock-duration 365
```

Sortie :

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d",
    "LockState": "compliance-cooloff",
    "LockDuration": 365,
    "CoolOffPeriod": 24,
    "CoolOffPeriodExpiresOn": "2024-05-06T01:02:20.527000+00:00",
    "LockCreatedOn": "2024-05-05T01:02:20.527000+00:00",
    "LockExpiresOn": "2025-05-05T01:02:20.527000+00:00",
    "LockDurationStartTime": "2024-05-05T01:02:20.527000+00:00"
}
```

Pour plus d'informations, consultez Snapshot Lock dans le guide de l'utilisateur d'Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous LockSnapshotà la section Référence des AWS CLI commandes.

`modify-address-attribute`

L'exemple de code suivant montre comment utiliser`modify-address-attribute`.

AWS CLI

Pour modifier l'attribut de nom de domaine associé à une adresse IP élastique

Les `modify-address-attribute` exemples suivants modifient l'attribut de nom de domaine d'une adresse IP élastique.

Linux :

```
aws ec2 modify-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --domain-name example.com
```

Windows :

```
aws ec2 modify-address-attribute ^
```

```
    --allocation-id eipalloc-abcdef01234567890 ^
    --domain-name example.com
```

Sortie :

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net."
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
        }
    ]
}
```

Pour surveiller le changement en attente et consulter les attributs modifiés d'une adresse IP élastique, consultez le manuel de référence describe-addresses-attributedes commandes de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous ModifyAddressAttributeà la section Référence des AWS CLI commandes.

`modify-availability-zone-group`

L'exemple de code suivant montre comment utilisermodify-availability-zone-group.

AWS CLI

Pour activer un groupe de zones

L'modify-availability-zone-groupexemple suivant active le groupe de zones spécifié.

```
aws ec2 modify-availability-zone-group \
    --group-name us-west-2-lax-1 \
    --opt-in-status opted-in
```

Sortie :

```
{
```

```
    "Return": true
}
```

Pour plus d'informations, consultez Régions et zones dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyAvailabilityZoneGroupà la section Référence des AWS CLI commandes.

`modify-capacity-reservation-fleet`

L'exemple de code suivant montre comment utiliser`modify-capacity-reservation-fleet`.

AWS CLI

Exemple 1 : pour modifier la capacité cible totale d'une flotte de réservation de capacité

L'`modify-capacity-reservation-fleet`exemple suivant modifie la capacité cible totale de la flotte de réservation de capacité spécifiée. Lorsque vous modifiez la capacité totale cible d'une flotte de réserve de capacité, la flotte crée automatiquement de nouvelles réserves de capacité, ou modifie ou annule les réserves de capacité existantes dans la flotte pour répondre à la nouvelle capacité totale cible. Vous ne pouvez pas tenter d'apporter d'autres modifications à une flotte lorsqu'elle se trouve dans l'état `modifying`.

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

Sortie :

```
{
    "Return": true
}
```

Exemple 2 : pour modifier la date de fin d'une flotte de réservation de capacité

L'`modify-capacity-reservation-fleet`exemple suivant modifie la date de fin de la flotte de réservation de capacité spécifiée. Lorsque vous modifiez la date de fin de la flotte, les dates de fin de toutes les réserves de capacité individuelles sont mises à jour en conséquence. Vous ne pouvez pas tenter d'apporter d'autres modifications à une flotte lorsqu'elle se trouve dans l'état `modifying`.

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2022-07-04T23:59:59.000Z
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations sur les flottes de réservation de capacité, consultez la section Flottes de réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyCapacityReservationFleetà la section Référence des AWS CLI commandes.

`modify-capacity-reservation`

L'exemple de code suivant montre comment utiliser`modify-capacity-reservation`.

AWS CLI

Exemple 1 : pour modifier le nombre d'instances réservées par une réservation de capacité existante

L'`modify-capacity-reservation`exemple suivant modifie le nombre d'instances pour lesquelles la réservation de capacité réserve de la capacité.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --instance-count 5
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez Modifier une réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour modifier la date et l'heure de fin d'une réservation de capacité existante

L'`modify-capacity-reservation`exemple suivant modifie une réservation de capacité existante pour qu'elle se termine à la date et à l'heure spécifiées.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --end-date-type limited \
    --end-date 2019-08-31T23:59:59Z
```

Pour plus d'informations, consultez Modifier une réservation de capacité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyCapacityReservationà la section Référence des AWS CLI commandes.

`modify-client-vpn-endpoint`

L'exemple de code suivant montre comment utiliser`modify-client-vpn-endpoint`.

AWS CLI

Pour modifier le point de terminaison d'un Client VPN

L'`modify-client-vpn-endpoint`exemple suivant active la journalisation des connexions client pour le point de terminaison VPN client spécifié.

```
aws ec2 modify-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez la section Points de terminaison du VPN client dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous ModifyClientVpnEndpointà la section Référence des AWS CLI commandes.

`modify-default-credit-specification`

L'exemple de code suivant montre comment utilisermodify-default-credit-specification.

AWS CLI

Pour modifier l'option de crédit par défaut

L'modify-default-credit-specificationexemple suivant modifie l'option de crédit par défaut pour les instances T2.

```
aws ec2 modify-default-credit-specification \
    --instance-family t2 \
    --cpu-credits unlimited
```

Sortie :

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "unlimited"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyDefaultCreditSpecificationà la section Référence des AWS CLI commandes.

`modify-ebs-default-kms-key-id`

L'exemple de code suivant montre comment utilisermodify-ebs-default-kms-key-id.

AWS CLI

Pour définir votre clé CMK par défaut pour le chiffrement EBS

L'modify-ebs-default-kms-key-idexemple suivant définit la clé CMK spécifiée comme clé CMK par défaut pour le chiffrement EBS de votre AWS compte dans la région actuelle.

```
aws ec2 modify-ebs-default-kms-key-id \
    --kms-key-id alias/my-cmk
```

Sortie :

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyEbsDefaultKmsKeyIdà la section Référence des AWS CLI commandes.

`modify-fleet`

L'exemple de code suivant montre comment utiliser`modify-fleet`.

AWS CLI

Pour faire évoluer une EC2 flotte

L'`modify-fleet`exemple suivant modifie la capacité cible de la EC2 flotte spécifiée. Si la valeur spécifiée est supérieure à la capacité actuelle, la EC2 flotte lance des instances supplémentaires. Si la valeur spécifiée est inférieure à la capacité actuelle, la EC2 flotte annule toutes les demandes ouvertes et si la politique de résiliation est la même`terminate`, la EC2 flotte met fin à toutes les instances qui dépassent la nouvelle capacité cible.

```
aws ec2 modify-fleet \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --target-capacity-specification TotalTargetCapacity=5
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez Gérer une EC2 flotte dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyFleetà la section Référence des AWS CLI commandes.

`modify-fpga-image-attribute`

L'exemple de code suivant montre comment utiliser`modify-fpga-image-attribute`.

AWS CLI

Pour modifier les attributs d'une image Amazon FPGA

Cet exemple ajoute des autorisations de chargement pour l'ID de compte `123456789012` pour l'AFI spécifié.

Commande :

```
aws ec2 modify-fpga-image-attribute --attribute loadPermission --fpga-image-
id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

Sortie :

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
          }
      ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyFpgaImageAttributeà la section Référence des AWS CLI commandes.

`modify-hosts`

L'exemple de code suivant montre comment utiliser`modify-hosts`.

AWS CLI

Exemple 1 : pour activer le placement automatique pour un hôte dédié

L'`modify-hosts`exemple suivant active le placement automatique d'un hôte dédié afin qu'il accepte tout lancement d'instance non ciblé correspondant à sa configuration de type d'instance.

```
aws ec2 modify-hosts \
    --host-id h-06c2f189b4EXAMPLE \
    --auto-placement on
```

Sortie :

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
}
```

Pour plus d'informations, consultez Modifier le paramètre de placement automatique pour un hôte dédié dans le guide de l' EC2 utilisateur Amazon.

Exemple 2 : Pour activer la restauration d'un hôte dédié

L'`modify-hosts`exemple suivant active la restauration de l'hôte pour l'hôte dédié spécifié.

```
aws ec2 modify-hosts \
    --host-id h-06c2f189b4EXAMPLE \
    --host-recovery on
```

Sortie :

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
}
```

Pour plus d'informations, consultez Modifier le paramètre de placement automatique pour un hôte dédié dans le guide de l' EC2 utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyHostsà la section Référence des AWS CLI commandes.

`modify-id-format`

L'exemple de code suivant montre comment utiliser`modify-id-format`.

AWS CLI

Pour activer le format d'identifiant plus long pour une ressource

L'`modify-id-format`exemple suivant active le format d'identifiant plus long pour le type de `instance` ressource.

```
aws ec2 modify-id-format \
    --resource instance \
    --use-long-ids
```

Pour désactiver le format d'identifiant plus long pour une ressource

L'`modify-id-format`exemple suivant désactive le format d'identifiant plus long pour le type de `instance` ressource.

```
aws ec2 modify-id-format \
    --resource instance \
    --no-use-long-ids
```

L'`modify-id-format`exemple suivant active le format d'identifiant plus long pour tous les types de ressources pris en charge dont la période d'inscription est en cours.

```
aws ec2 modify-id-format \
    --resource all-current \
    --use-long-ids
```

- Pour plus de détails sur l'API, reportez-vous ModifyIdFormatà la section Référence des AWS CLI commandes.

`modify-identity-id-format`

L'exemple de code suivant montre comment utiliser`modify-identity-id-format`.

AWS CLI

Pour permettre à un rôle IAM d'utiliser une ressource plus IDs longtemps

L'`modify-identity-id-format`exemple suivant permet au rôle `EC2Role` IAM de votre AWS compte d'utiliser le format d'identifiant long pour le type de `instance` ressource.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/EC2Role \
    --resource instance \
    --use-long-ids
```

Pour permettre à un utilisateur IAM d'utiliser une ressource plus IDs longtemps

L'`modify-identity-id-format`exemple suivant permet à l'utilisateur `AdminUser` IAM de votre AWS compte d'utiliser le format d'identifiant plus long pour le type de `volume` ressource.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource volume \
    --use-long-ids
```

L'`modify-identity-id-format`exemple suivant permet à l'utilisateur `AdminUser` IAM de votre AWS compte d'utiliser le format d'identifiant plus long pour tous les types de ressources pris en charge pendant leur période d'inscription.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource all-current \
    --use-long-ids
```

- Pour plus de détails sur l'API, reportez-vous ModifyIdentityIdFormatà la section Référence des AWS CLI commandes.

`modify-image-attribute`

L'exemple de code suivant montre comment utiliser`modify-image-attribute`.

AWS CLI

Exemple 1 : Pour rendre publique une AMI

L'`modify-instance-attribute`exemple suivant rend publique l'AMI spécifiée.

```
aws ec2 modify-image-attribute \
```

```
    --image-id ami-5731123e \
    --launch-permission "Add=[{Group=all}]"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour rendre une AMI privée

L'`modify-instance-attribute`exemple suivant rend l'AMI spécifiée privée.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{Group=all}]"
```

Cette commande ne produit aucun résultat.

Exemple 3 : accorder une autorisation de lancement à un AWS compte

L'`modify-instance-attribute`exemple suivant accorde des autorisations de lancement au AWS compte spécifié.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{UserId=123456789012}]"
```

Cette commande ne produit aucun résultat.

Exemple 4 : pour supprimer l'autorisation de lancement d'un AWS compte

L'`modify-instance-attribute`exemple suivant supprime les autorisations de lancement du AWS compte spécifié.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

- Pour plus de détails sur l'API, reportez-vous ModifyImageAttributeà la section Référence des AWS CLI commandes.

`modify-instance-attribute`

L'exemple de code suivant montre comment utiliser`modify-instance-attribute`.

AWS CLI

Exemple 1 : pour modifier le type d'instance

L'`modify-instance-attribute`exemple suivant modifie le type d'instance de l'instance spécifiée. L'instance doit être dans l'état `stopped`.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --instance-type "{\"Value\": \"m1.small\"}"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour activer la mise en réseau améliorée sur une instance

L'`modify-instance-attribute`exemple suivant permet d'améliorer la mise en réseau pour l'instance spécifiée. L'instance doit être dans l'état `stopped`.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

Cette commande ne produit aucun résultat.

Exemple 3 : pour modifier l' sourceDestCheck attribut

L'`modify-instance-attribute`exemple suivant définit l'`sourceDestCheck`attribut de l'instance spécifiée sur`true`. L'instance doit se trouver dans un VPC.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-
check "{\"Value\": true}"
```

Cette commande ne produit aucun résultat.

Exemple 4 : pour modifier l' deleteOnTermination attribut du volume racine

L'`modify-instance-attribute`exemple suivant définit l'`deleteOnTermination`attribut du volume racine de l'instance basée sur Amazon EBS spécifiée sur. `false` Par défaut, cet attribut est `true` destiné au volume racine.

Commande :

```
aws ec2 modify-instance-attribute \
```

```
  --instance-id i-1234567890abcdef0 \
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\",\"Ebs\":
{\"DeleteOnTermination\":false}}]"
```

Cette commande ne produit aucun résultat.

Exemple 5 : pour modifier les données utilisateur associées à une instance

L'`modify-instance-attribute`exemple suivant ajoute le contenu du fichier `UserData.txt` tel que celui UserData de l'instance spécifiée.

Contenu du fichier original `UserData.txt` :

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

Le contenu du fichier doit être codé en base64. La première commande convertit le fichier texte en base64 et l'enregistre en tant que nouveau fichier.

Version Linux/macOS de la commande :

```
base64 UserData.txt > UserData.base64.txt
```

Cette commande ne produit aucun résultat.

Version Windows de la commande :

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >
 UserData.base64.txt
```

Sortie :

```
Input Length = 67
Output Length = 152
CertUtil: -encode command completed successfully.
```

Vous pouvez maintenant référencer ce fichier dans la commande CLI suivante :

```
aws ec2 modify-instance-attribute \
```

```
    --instance-id=i-09b5a14dbca622e76 \
    --attribute userData --value file://UserData.base64.txt
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Données utilisateur et AWS CLI dans le guide de EC2 l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceAttributeà la section Référence des AWS CLI commandes.

`modify-instance-capacity-reservation-attributes`

L'exemple de code suivant montre comment utiliser`modify-instance-capacity-reservation-attributes`.

AWS CLI

Exemple 1 : pour modifier les paramètres de ciblage des réservations de capacité d'une instance

L'`modify-instance-capacity-reservation-attributes`exemple suivant modifie une instance arrêtée pour cibler une réservation de capacité spécifique.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification
 'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez Modifier les paramètres de réservation de capacité de votre instance dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour modifier les paramètres de ciblage des réservations de capacité d'une instance

L'`modify-instance-capacity-reservation-attributes`exemple suivant modifie une instance arrêtée qui cible la réservation de capacité spécifiée pour qu'elle soit lancée dans toute

réservation de capacité dont les attributs correspondent (type d'instance, plateforme, zone de disponibilité) et dont les critères de correspondance d'instance sont ouverts.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification 'CapacityReservationPreference=open'
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez Modifier les paramètres de réservation de capacité de votre instance dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceCapacityReservationAttributesà la section Référence des AWS CLI commandes.

`modify-instance-credit-specification`

L'exemple de code suivant montre comment utiliser`modify-instance-credit-specification`.

AWS CLI

Pour modifier l'option de crédit pour l'utilisation du processeur d'une instance

Cet exemple modifie l'option de crédit pour l'utilisation du processeur de l'instance spécifiée dans la région spécifiée en « illimitée ». Les options de crédit valides sont « standard » et « illimitées ».

Commande :

```
aws ec2 modify-instance-credit-specification --instance-credit-
specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

Sortie :

```
{
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0"
```

```
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceCreditSpecificationà la section Référence des AWS CLI commandes.

`modify-instance-event-start-time`

L'exemple de code suivant montre comment utilisermodify-instance-event-start-time.

AWS CLI

Pour modifier l'heure de début de l'événement pour une instance

La `modify-instance-event-start-time` commande suivante montre comment modifier l'heure de début de l'événement pour l'instance spécifiée. Spécifiez l'ID de l'événement à l'aide du `--instance-event-id` paramètre. Spécifiez la nouvelle date et heure à l'aide du `--not-before` paramètre.

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0
 --instance-event-id instance-event-0abcdef1234567890 --not-
before 2019-03-25T10:00:00.000
```

Sortie :

```
"Event": {
    "InstanceEventId": "instance-event-0abcdef1234567890",
    "Code": "system-reboot",
    "Description": "scheduled reboot",
    "NotAfter": "2019-03-25T12:00:00.000Z",
    "NotBefore": "2019-03-25T10:00:00.000Z",
    "NotBeforeDeadline": "2019-04-22T21:00:00.000Z"
}
```

Pour plus d'informations, consultez la section Travailler avec des instances planifiées pour le redémarrage dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceEventStartTimeà la section Référence des AWS CLI commandes.

`modify-instance-event-window`

L'exemple de code suivant montre comment utiliser`modify-instance-event-window`.

AWS CLI

Exemple 1 : pour modifier la plage de temps d'une fenêtre d'événements

L'`modify-instance-event-window`exemple suivant modifie la plage de temps d'une fenêtre d'événements. Spécifiez le paramètre `time-range` pour modifier la plage de temps. Vous ne pouvez pas également spécifier le paramètre `cron-expression`.

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
```

```
                "Value": "V1"
            }
        ]
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour modifier un ensemble de plages de temps pour une fenêtre d'événements

L'`modify-instance-event-window`exemple suivant modifie la plage de temps d'une fenêtre d'événements. Spécifiez le paramètre `time-range` pour modifier la plage de temps. Vous ne pouvez pas également spécifier le paramètre `cron-expression`.

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
 "wednesday", "EndHour": 8},
        {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
 "EndHour": 8}]'
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            },
            {
                "StartWeekDay": "thursday",
                "StartHour": 2,
                "EndWeekDay": "friday",
                "EndHour": 8
            }
        ],
```

```
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

Exemple 3 : pour modifier l'expression cron d'une fenêtre d'événements

L'`modify-instance-event-window`exemple suivant modifie l'expression cron d'une fenêtre d'événements. Spécifiez le paramètre `cron-expression` pour modifier l'expression cron. Vous ne pouvez pas également spécifier le paramètre `time-range`.

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

Sortie :

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
```

```
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

Pour les contraintes liées à la fenêtre d'événements, consultez les considérations dans la section Événements planifiés du guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceEventWindowà la section Référence des AWS CLI commandes.

`modify-instance-maintenance-options`

L'exemple de code suivant montre comment utiliser`modify-instance-maintenance-options`.

AWS CLI

Exemple 1 : pour désactiver le comportement de restauration d'une instance

L'`modify-instance-maintenance-options`exemple suivant désactive la restauration automatique simplifiée pour une instance en cours d'exécution ou arrêtée.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-0abcdef1234567890 \
    --auto-recovery disabled
```

Sortie :

```
{
    "InstanceId": "i-0abcdef1234567890",
    "AutoRecovery": "disabled"
}
```

Pour plus d'informations, consultez Configurer la restauration automatique simplifiée dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour définir le comportement de restauration d'une instance par défaut

L'`modify-instance-maintenance-options`exemple suivant définit le comportement de restauration automatique par défaut, ce qui permet une restauration automatique simplifiée pour les types d'instances pris en charge.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-0abcdef1234567890 \
    --auto-recovery default
```

Sortie :

```
{
    "InstanceId": "i-0abcdef1234567890",
    "AutoRecovery": "default"
}
```

Pour plus d'informations, consultez Configurer la restauration automatique simplifiée dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceMaintenanceOptionsà la section Référence des AWS CLI commandes.

`modify-instance-metadata-options`

L'exemple de code suivant montre comment utiliser`modify-instance-metadata-options`.

AWS CLI

Exemple 1 : pour activer IMDSv2

L'`modify-instance-metadata-options`exemple suivant configure l'utilisation de IMDSv2 sur l'instance spécifiée.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

Sortie :

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
}
```

Pour plus d'informations, consultez la section Métadonnées de l'instance dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour désactiver les métadonnées de l'instance

L'`modify-instance-metadata-options`exemple suivant désactive l'utilisation de toutes les versions des métadonnées d'instance sur l'instance spécifiée.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-endpoint disabled
```

Sortie :

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "disabled"
    }
}
```

Pour plus d'informations, consultez la section Métadonnées de l'instance dans le guide de EC2 l'utilisateur Amazon.

Exemple 3 : Pour activer le point de IPv6 terminaison des métadonnées d'instance pour votre instance

L'`modify-instance-metadata-options`exemple suivant montre comment activer le point de IPv6 terminaison pour le service de métadonnées d'instance. Par défaut, le IPv6 point de terminaison est désactivé. Cela est vrai même si vous avez lancé une instance dans un IPv6 sous-réseau uniquement. Le IPv6 point de terminaison pour IMDS n'est accessible que sur les instances créées sur le système Nitro.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-protocol-ipv6 enabled \
    --http-endpoint enabled
```

Sortie :

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled",
        HttpProtocolIpv6": "enabled"
    }
}
```

Pour plus d'informations, consultez la section Métadonnées de l'instance dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceMetadataOptionsà la section Référence des AWS CLI commandes.

`modify-instance-placement`

L'exemple de code suivant montre comment utiliser`modify-instance-placement`.

AWS CLI

Exemple 1 : pour supprimer l'affinité d'une instance avec un hôte dédié

L'`modify-instance-placement`exemple suivant supprime l'affinité d'une instance avec un hôte dédié et lui permet de se lancer sur n'importe quel hôte dédié disponible dans votre compte qui prend en charge son type d'instance.

```
aws ec2 modify-instance-placement \
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity default
```

Sortie :

```
{
    "Return": true
}
```

Exemple 2 : pour établir une affinité entre une instance et l'hôte dédié spécifié

L'`modify-instance-placement`exemple suivant établit une relation de lancement entre une instance et un hôte dédié. L'instance ne peut s'exécuter que sur l'hôte dédié spécifié.

```
aws ec2 modify-instance-placement \
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity host \
    --host-id i-0e6ddf6187EXAMPLE
```

Sortie :

```
{
    "Return": true
}
```

Exemple 3 : pour déplacer une instance vers un groupe de placement

L'`modify-instance-placement`exemple suivant déplace une instance vers un groupe de placement, arrête l'instance, modifie le placement de l'instance, puis redémarre l'instance.

```
aws ec2 stop-instances \
    --instance-ids i-0123a456700123456

aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name MySpreadGroup

aws ec2 start-instances \
    --instance-ids i-0123a456700123456
```

Exemple 4 : pour supprimer une instance d'un groupe de placement

L'`modify-instance-placement`exemple suivant supprime une instance d'un groupe de placement en arrêtant l'instance, en modifiant le placement de l'instance, puis en la redémarrant. L'exemple suivant spécifie une chaîne vide (« ») pour le nom du groupe de placement afin d'indiquer que l'instance ne doit pas être située dans un groupe de placement.

Arrêtez l'instance :

```
aws ec2 stop-instances \
    --instance-ids i-0123a456700123456
```

Modifiez le placement (invite de commandes Windows) :

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

Modifiez l'emplacement (Windows PowerShell, Linux et macOS) :

```
aws ec2 modify-instance-placement `
    --instance-id i-0123a456700123456 `
    --group-name ''
```

Redémarrez l'instance :

```
aws ec2 start-instances \
    --instance-ids i-0123a456700123456
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez Modifier la location et l'affinité d'un hôte dédié dans le guide de l' EC2 utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyInstancePlacementà la section Référence des AWS CLI commandes.

`modify-ipam-pool`

L'exemple de code suivant montre comment utiliser`modify-ipam-pool`.

AWS CLI

Pour modifier un pool IPAM

L'`modify-ipam-pool`exemple suivant modifie un pool IPAM.

(Linux) :

```
aws ec2 modify-ipam-pool \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --add-allocation-resource-tags "Key=Owner,Value=Build Team" \
    --clear-allocation-default-netmask-length \
    --allocation-min-netmask-length 14
```

(Fenêtres) :

```
aws ec2 modify-ipam-pool ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --add-allocation-resource-tags "Key=Owner,Value=Build Team" ^
    --clear-allocation-default-netmask-length ^
    --allocation-min-netmask-length 14
```

Sortie :

```
{
    "IpamPool": {
        "OwnerId": "123456789012",
        "IpamPoolId": "ipam-pool-0533048da7d823723",
        "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-0533048da7d823723",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
        "IpamScopeType": "private",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "Locale": "None",
        "PoolDepth": 1,
        "State": "modify-complete",
```

```
        "AutoImport": true,
        "AddressFamily": "ipv4",
        "AllocationMinNetmaskLength": 14,
        "AllocationMaxNetmaskLength": 26,
        "AllocationResourceTags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            },
            {
                "Key": "Owner",
                "Value": "Build Team"
            }
        ]
    }
}
```

Pour plus d'informations, consultez Modifier un pool dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous ModifyIpamPoolà la section Référence des AWS CLI commandes.

`modify-ipam-resource-cidr`

L'exemple de code suivant montre comment utiliser`modify-ipam-resource-cidr`.

AWS CLI

Pour modifier le CIDR attribué à une ressource

L'`modify-ipam-resource-cidr`exemple suivant modifie un CIDR de ressource.

(Linux) :

```
aws ec2 modify-ipam-resource-cidr \
    --current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
    --destination-ipam-scope-id ipam-scope-0da34c61fd189a141 \
    --resource-id vpc-010e1791024eb0af9 \
    --resource-cidr 10.0.1.0/24 \
    --resource-region us-east-1 \
    --monitored
```

(Fenêtres) :

```
aws ec2 modify-ipam-resource-cidr ^
    --current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
    --destination-ipam-scope-id ipam-scope-0da34c61fd189a141 ^
    --resource-id vpc-010e1791024eb0af9 ^
    --resource-cidr 10.0.1.0/24 ^
    --resource-region us-east-1 ^
    --monitored
```

Sortie :

```
{
    "IpamResourceCidr": {
        "IpamId": "ipam-08440e7a3acde3908",
        "IpamScopeId": "ipam-scope-0da34c61fd189a141",
        "IpamPoolId": "ipam-pool-0533048da7d823723",
        "ResourceRegion": "us-east-1",
        "ResourceOwnerId": "123456789012",
        "ResourceId": "vpc-010e1791024eb0af9",
        "ResourceCidr": "10.0.1.0/24",
        "ResourceType": "vpc",
        "ResourceTags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            },
            {
                "Key": "Owner",
                "Value": "Build Team"
            }
        ],
        "IpUsage": 0.0,
        "ComplianceStatus": "noncompliant",
        "ManagementState": "managed",
        "OverlapStatus": "overlapping",
        "VpcId": "vpc-010e1791024eb0af9"
    }
}
```

Pour plus d'informations sur le déplacement de ressources, consultez la section Déplacer des ressources CIDRs entre des étendues dans le guide de l'utilisateur Amazon VPC IPAM.

Pour plus d'informations sur la modification des états de surveillance, consultez Modifier l'état de surveillance de la ressource CIDRs dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous ModifyIpamResourceCidrà la section Référence des AWS CLI commandes.

`modify-ipam-resource-discovery`

L'exemple de code suivant montre comment utiliser`modify-ipam-resource-discovery`.

AWS CLI

Pour modifier les régions opérationnelles d'une découverte de ressources

Dans cet exemple, vous êtes un administrateur délégué IPAM qui souhaite modifier les régions opérationnelles d'une découverte de ressources.

Pour compléter cette demande :

Vous ne pouvez pas modifier une découverte de ressource par défaut et vous devez en être le propriétaire. Vous avez besoin de l'ID de découverte de ressources, que vous pouvez obtenir. describe-ipam-resource-discoveries

L'`modify-ipam-resource-discovery`exemple suivant modifie une découverte de ressources autre que celle par défaut dans votre AWS compte.

```
aws ec2 modify-ipam-resource-discovery \
    --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
    --add-operating-regions RegionName='us-west-1' \
    --remove-operating-regions RegionName='us-east-2' \
    --region us-east-1
```

Sortie :

```
{
    "IpamResourceDiscovery": {
        "OwnerId": "149977607591",
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
        "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0365d2977fc1672fe",
        "IpamResourceDiscoveryRegion": "us-east-1",
        "Description": "Example",
```

```
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            },
            {
                "RegionName": "us-west-1"
            }
        ],
        "IsDefault": false,
        "State": "modify-in-progress"
    }
}
```

Pour plus d'informations, consultez la section Travailler avec les découvertes de ressources dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous ModifyIpamResourceDiscoveryà la section Référence des AWS CLI commandes.

`modify-ipam-scope`

L'exemple de code suivant montre comment utiliser`modify-ipam-scope`.

AWS CLI

Pour modifier la description d'une étendue

Dans ce scénario, vous êtes un administrateur délégué IPAM qui souhaite modifier la description d'une étendue IPAM.

Pour compléter cette demande, vous aurez besoin de l'identifiant de portée, que vous pourrez obtenir describe-ipam-scopes.

L'`modify-ipam-scope`exemple suivant met à jour la description de la portée.

```
aws ec2 modify-ipam-scope \
    --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \
    --description example \
    --region us-east-1
```

Sortie :

```
{
```

```
"IpamScope": {
        "OwnerId": "320805250157",
        "IpamScopeId": "ipam-scope-0d3539a30b57dcdd1",
        "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0d3539a30b57dcdd1",
        "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
        "IpamRegion": "us-east-1",
        "IpamScopeType": "public",
        "IsDefault": true,
        "Description": "example",
        "PoolCount": 1,
        "State": "modify-in-progress"
    }
}
```

Pour plus d'informations sur les étendues, consultez Comment fonctionne l'IPAM dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous ModifyIpamScopeà la section Référence des AWS CLI commandes.

`modify-ipam`

L'exemple de code suivant montre comment utiliser`modify-ipam`.

AWS CLI

Pour modifier un IPAM

L'`modify-ipam`exemple suivant modifie un IPAM en ajoutant une région opérationnelle.

(Linux) :

```
aws ec2 modify-ipam \
    --ipam-id ipam-08440e7a3acde3908 \
    --add-operating-regions RegionName=us-west-2
```

(Fenêtres) :

```
aws ec2 modify-ipam ^
    --ipam-id ipam-08440e7a3acde3908 ^
    --add-operating-regions RegionName=us-west-2
```

Sortie :

```
{
    "Ipam": {
        "OwnerId": "123456789012",
        "IpamId": "ipam-08440e7a3acde3908",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
        "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
        "ScopeCount": 3,
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            },
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-west-1"
            },
            {
                "RegionName": "us-west-2"
            }
        ],
        "State": "modify-in-progress"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyIpamà la section Référence des AWS CLI commandes.

`modify-launch-template`

L'exemple de code suivant montre comment utiliser`modify-launch-template`.

AWS CLI

Pour modifier la version du modèle de lancement par défaut

Cet exemple indique la version 2 du modèle de lancement spécifié comme version par défaut.

Commande :

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-
version 2
```

Sortie :

```
{
  "LaunchTemplate": {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0abcd290751193123",
      "LaunchTemplateName": "WebServers",
      "DefaultVersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyLaunchTemplateà la section Référence des AWS CLI commandes.

`modify-managed-prefix-list`

L'exemple de code suivant montre comment utiliser`modify-managed-prefix-list`.

AWS CLI

Pour modifier une liste de préfixes

L'`modify-managed-prefix-list`exemple suivant ajoute une entrée à la liste de préfixes spécifiée.

```
aws ec2 modify-managed-prefix-list \
    --prefix-list-id pl-0123456abcabcabc1 \
    --add-entries Cidr=10.1.0.0/16,Description=vpc-c \
    --current-version 1
```

Sortie :

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
```

```
        "State": "modify-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 1,
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous ModifyManagedPrefixListà la section Référence des AWS CLI commandes.

`modify-network-interface-attribute`

L'exemple de code suivant montre comment utiliser`modify-network-interface-attribute`.

AWS CLI

Pour modifier l'attribut de pièce jointe d'une interface réseau

Cet exemple de commande modifie l'`attachment`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

Pour modifier l'attribut de description d'une interface réseau

Cet exemple de commande modifie l'`description`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
description "My description"
```

Pour modifier l'attribut GroupSet d'une interface réseau

Cet exemple de commande modifie l'`groupSet`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
groups sg-903004f8 sg-1a2b3c4d
```

Pour modifier l' sourceDestCheck attribut d'une interface réseau

Cet exemple de commande modifie l'`sourceDestCheck`attribut de l'interface réseau spécifiée.

Commande :

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-
source-dest-check
```

- Pour plus de détails sur l'API, reportez-vous ModifyNetworkInterfaceAttributeà la section Référence des AWS CLI commandes.

`modify-private-dns-name-options`

L'exemple de code suivant montre comment utiliser`modify-private-dns-name-options`.

AWS CLI

Pour modifier les options relatives aux noms d'hôte par exemple

L'`modify-private-dns-name-options`exemple suivant désactive l'option permettant de répondre aux requêtes DNS, par exemple, aux noms d'hôte contenant des enregistrements DNS A.

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --no-enable-resource-name-dns-a-record
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez les types de noms d'hôte des EC2 instances Amazon dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyPrivateDnsNameOptionsà la section Référence des AWS CLI commandes.

`modify-reserved-instances`

L'exemple de code suivant montre comment utiliser`modify-reserved-instances`.

AWS CLI

Pour modifier les instances réservées

Cet exemple de commande déplace une instance réservée vers une autre zone de disponibilité de la même région.

Commande :

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=10
```

Sortie :

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

Pour modifier la plate-forme réseau des instances réservées

Cet exemple de commande convertit EC2 -Classic Reserved Instances en EC2 -VPC.

Commande :

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-VPC,InstanceCount=5
```

Sortie :

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

Pour plus d'informations, consultez la section Modification de vos instances réservées dans le guide de EC2 l'utilisateur Amazon.

Pour modifier la taille d'instance des instances réservées

Cet exemple de commande modifie une instance réservée qui possède 10 instances m1.small dans us-west-1c afin que 8 Linux/UNIX instances m1.small deviennent 2 instances m1.large, et les 2 instances m1.small restantes deviennent 1 instance m1.medium dans la même zone de disponibilité. Commande :

```
aws ec2 modify-reserved-instances --reserved-instances-
ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-
configurations AvailabilityZone=us-west-1c,Platform=EC2-
Classic,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-
west-1c,Platform=EC2-Classic,InstanceCount=1,InstanceType=m1.medium
```

Sortie :

```
{
    "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

Pour plus d'informations, consultez la section Modification de la taille de l'instance de vos réservations dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifyReservedInstancesà la section Référence des AWS CLI commandes.

`modify-security-group-rules`

L'exemple de code suivant montre comment utiliser`modify-security-group-rules`.

AWS CLI

Pour modifier les règles d'un groupe de sécurité afin de mettre à jour la description de la règle, le protocole IP et la plage d'adresses CidrIpv 4

L'`modify-security-group-rules`exemple suivant met à jour la description, le protocole IP et la plage d'adresses IPV4 CIDR d'une règle de groupe de sécurité spécifiée. Utilisez le `security-group-rules` paramètre pour saisir les mises à jour des règles du groupe de sécurité spécifiées. `-1`spécifie tous les protocoles.

```
aws ec2 modify-security-group-rules \
    --group-id sg-1234567890abcdef0 \
    --security-group-rules SecurityGroupRuleId=sgr-
abcdef01234567890,SecurityGroupRule='{Description=test,IpProtocol=-1,CidrIpv4=0.0.0.0/0}'
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations sur les règles des groupes de sécurité, consultez la section Règles des groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ModifySecurityGroupRulesà la section Référence des AWS CLI commandes.

`modify-snapshot-attribute`

L'exemple de code suivant montre comment utiliser`modify-snapshot-attribute`.

AWS CLI

Exemple 1 : pour modifier un attribut de capture d'écran

L'`modify-snapshot-attribute`exemple suivant met à jour l'`createVolumePermission`attribut pour le cliché spécifié, en supprimant les autorisations de volume pour l'utilisateur spécifié.

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type remove \
    --user-ids 123456789012
```

Exemple 2 : Pour rendre public un instantané

L'`modify-snapshot-attribute`exemple suivant rend public le cliché spécifié.

```
aws ec2 modify-snapshot-attribute \
```

```
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type add \
    --group-names all
```

- Pour plus de détails sur l'API, reportez-vous ModifySnapshotAttributeà la section Référence des AWS CLI commandes.

`modify-snapshot-tier`

L'exemple de code suivant montre comment utiliser`modify-snapshot-tier`.

AWS CLI

Pour archiver un instantané

L'`modify-snapshot-tier`exemple suivant archive le cliché spécifié. Le paramètre de réponse `TieringStartTime` indique la date et l'heure à laquelle le processus d'archivage a été lancé, au format UTC (AAAA-MM-JJTHH:MM:SSZ).

```
aws ec2 modify-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --storage-tier archive
```

Sortie :

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

Pour plus d'informations sur l'archivage des instantanés, consultez la section Archiver des instantanés Amazon EBS dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous ModifySnapshotTierà la section Référence des AWS CLI commandes.

`modify-spot-fleet-request`

L'exemple de code suivant montre comment utiliser`modify-spot-fleet-request`.

AWS CLI

Pour modifier une demande de flotte Spot

Cet exemple de commande met à jour la capacité cible de la demande de parc Spot spécifiée.

Commande :

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-
id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Sortie :

```
{
    "Return": true
}
```

Cet exemple de commande réduit la capacité cible de la demande de parc Spot spécifiée sans mettre fin à aucune instance Spot en conséquence.

Commande :

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-
termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE
```

Sortie :

```
{
    "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous ModifySpotFleetRequestà la section Référence des AWS CLI commandes.

`modify-subnet-attribute`

L'exemple de code suivant montre comment utiliser`modify-subnet-attribute`.

AWS CLI

Pour modifier le comportement d' IPv4 adressage public d'un sous-réseau

Cet exemple modifie le sous-réseau -1a2b3c4d pour spécifier qu'une adresse publique est attribuée à toutes les instances lancées dans ce sous-réseau. IPv4 Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-launch
```

Pour modifier le comportement d' IPv6 adressage d'un sous-réseau

Cet exemple modifie le sous-réseau -1a2b3c4d pour spécifier que toutes les instances lancées dans ce sous-réseau se voient attribuer une adresse provenant de la plage du sous-réseau. IPv6

Commande :

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

Pour plus d'informations, consultez la section Adressage IP dans votre VPC dans le Guide de l'utilisateur du cloud privéAWS virtuel.

- Pour plus de détails sur l'API, reportez-vous ModifySubnetAttributeà la section Référence des AWS CLI commandes.

`modify-traffic-mirror-filter-network-services`

L'exemple de code suivant montre comment utiliser`modify-traffic-mirror-filter-network-services`.

AWS CLI

Pour ajouter des services réseau à un filtre Traffic Mirror

L'`modify-traffic-mirror-filter-network-services`exemple suivant ajoute les services réseau Amazon DNS au filtre spécifié.

```
aws ec2 modify-traffic-mirror-filter-network-services \
    --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \
    --add-network-service amazon-dns
```

Sortie :

```
{
    "TrafficMirrorFilter": {
        "Tags": [
            {
                "Key": "Name",
                "Value": "Production"
            }
        ],
        "EgressFilterRules": [],
        "NetworkServices": [
            "amazon-dns"
        ],
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "IngressFilterRules": [
            {
                "SourceCidrBlock": "0.0.0.0/0",
                "RuleNumber": 1,
                "DestinationCidrBlock": "0.0.0.0/0",
                "Description": "TCP Rule",
                "Protocol": 6,
                "TrafficDirection": "ingress",
                "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
                "RuleAction": "accept",
                "TrafficMirrorFilterRuleId": "tmf-04812ff784EXAMPLE"
            }
        ]
    }
}
```

Pour plus d'informations, voir Modifier les services réseau du filtre Traffic Mirror dans le Guide de mise en miroir AWS du trafic.

- Pour plus de détails sur l'API, reportez-vous ModifyTrafficMirrorFilterNetworkServicesà la section Référence des AWS CLI commandes.

`modify-traffic-mirror-filter-rule`

L'exemple de code suivant montre comment utiliser`modify-traffic-mirror-filter-rule`.

AWS CLI

Pour modifier une règle de filtrage du trafic miroir

L'`modify-traffic-mirror-filter-rule`exemple suivant modifie la description de la règle de filtrage du trafic spécifiée.

```
aws ec2 modify-traffic-mirror-filter-rule \
    --traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \
    --description "TCP Rule"
```

Sortie :

```
{
    "TrafficMirrorFilterRule": {
        "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
        "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
        "TrafficDirection": "ingress",
        "RuleNumber": 100,
        "RuleAction": "accept",
        "Protocol": 6,
        "DestinationCidrBlock": "10.0.0.0/24",
        "SourceCidrBlock": "10.0.0.0/24",
        "Description": "TCP Rule"
    }
}
```

Pour plus d'informations, consultez la section Modifier vos règles de filtrage du trafic dans le guide de mise en miroir AWS du trafic.

- Pour plus de détails sur l'API, reportez-vous ModifyTrafficMirrorFilterRuleà la section Référence des AWS CLI commandes.

`modify-traffic-mirror-session`

L'exemple de code suivant montre comment utiliser`modify-traffic-mirror-session`.

AWS CLI

Pour modifier une session de miroir du trafic

L'`modify-traffic-mirror-session`exemple suivant modifie la description de la session de miroir du trafic et le nombre de paquets à mettre en miroir.

```
aws ec2 modify-traffic-mirror-session \
    --description "Change packet length" \
```

```
    --traffic-mirror-session-id tms-08a33b1214EXAMPLE \
    --remove-fields "packet-length"
```

Sortie :

```
{
    "TrafficMirrorSession": {
        "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
        "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "NetworkInterfaceId": "eni-070203f901EXAMPLE",
        "OwnerId": "111122223333",
        "SessionNumber": 1,
        "VirtualNetworkId": 7159709,
        "Description": "Change packet length",
        "Tags": []
    }
}
```

Pour plus d'informations, voir Modifier votre session de miroir du trafic dans le Guide de mise en miroir du trafic.

- Pour plus de détails sur l'API, reportez-vous ModifyTrafficMirrorSessionà la section Référence des AWS CLI commandes.

`modify-transit-gateway-prefix-list-reference`

L'exemple de code suivant montre comment utilisermodify-transit-gateway-prefix-list-reference.

AWS CLI

Pour modifier une référence à une liste de préfixes

L'modify-transit-gateway-prefix-list-referenceexemple suivant modifie la référence de la liste de préfixes dans la table de routage spécifiée en modifiant la pièce jointe vers laquelle le trafic est acheminé.

```
aws ec2 modify-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-1111122222222333 \
    --transit-gateway-attachment-id tgw-attach-aabbccddaabbccaab
```

Sortie :

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "modifying",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

Pour plus d'informations, voir les références aux listes de préfixes dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous ModifyTransitGatewayPrefixListReferenceà la section Référence des AWS CLI commandes.

`modify-transit-gateway-vpc-attachment`

L'exemple de code suivant montre comment utiliser`modify-transit-gateway-vpc-attachment`.

AWS CLI

Pour modifier une pièce jointe VPC d'une passerelle de transit

L'`modify-transit-gateway-vpc-attachment`exemple suivant ajoute un sous-réseau à la pièce jointe VPC de passerelle de transit spécifiée.

```
aws ec2 modify-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
    --add-subnet-ids subnet-0e51f45802EXAMPLE
```

Sortie :

```
{
```

```
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-09fbd47ddfEXAMPLE",
        "TransitGatewayId": "tgw-0560315ccfEXAMPLE",
        "VpcId": "vpc-5eccc927",
        "VpcOwnerId": "111122223333",
        "State": "modifying",
        "SubnetIds": [
            "subnet-0e51f45802EXAMPLE",
            "subnet-1EXAMPLE"
        ],
        "CreationTime": "2019-08-08T16:47:38.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

Pour plus d'informations, consultez la section Pièces jointes d'une passerelle de transit à un VPC dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous ModifyTransitGatewayVpcAttachmentà la section Référence des AWS CLI commandes.

`modify-transit-gateway`

L'exemple de code suivant montre comment utiliser`modify-transit-gateway`.

AWS CLI

Pour modifier une passerelle de transit

L'`modify-transit-gateway`exemple suivant modifie la passerelle de transit spécifiée en activant le support ECMP pour les pièces jointes VPN.

```
aws ec2 modify-transit-gateway \
    --transit-gateway-id tgw-111111222222aaaaa \
    --options VpnEcmpSupport=enable
```

Sortie :

```
{
    "TransitGateway": {
```

```
        "TransitGatewayId": "tgw-111111222222aaaaa",
        "TransitGatewayArn": "64512",
        "State": "modifying",
        "OwnerId": "123456789012",
        "CreationTime": "2020-04-30T08:41:37.000Z",
        "Options": {
            "AmazonSideAsn": 64512,
            "AutoAcceptSharedAttachments": "disable",
            "DefaultRouteTableAssociation": "enable",
            "AssociationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
            "DefaultRouteTablePropagation": "enable",
            "PropagationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
            "VpnEcmpSupport": "enable",
            "DnsSupport": "enable"
        }
    }
}
```

Pour plus d'informations, consultez la section Passerelles de transport en commun dans le Guide des passerelles de transport en commun.

- Pour plus de détails sur l'API, reportez-vous ModifyTransitGatewayà la section Référence des AWS CLI commandes.

`modify-verified-access-endpoint-policy`

L'exemple de code suivant montre comment utiliser`modify-verified-access-endpoint-policy`.

AWS CLI

Pour configurer la politique d'accès vérifié pour un point de terminaison

L'`modify-verified-access-endpoint-policy`exemple suivant ajoute la politique d'accès vérifié spécifiée au point de terminaison d'accès vérifié spécifié.

```
aws ec2 modify-verified-access-endpoint-policy \
    --verified-access-endpoint-id vae-066fac616d4d546f2 \
    --policy-enabled \
    --policy-document file://policy.txt
```

Contenu de `policy.txt` :

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

Sortie :

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

Pour plus d'informations, consultez les politiques d'accès vérifié dans le guide de l'utilisateur de l'accès AWS vérifié.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessEndpointPolicyà la section Référence des AWS CLI commandes.

`modify-verified-access-endpoint`

L'exemple de code suivant montre comment utiliser`modify-verified-access-endpoint`.

AWS CLI

Pour modifier la configuration d'un point de terminaison d'accès vérifié

L'`modify-verified-access-endpoint`exemple suivant ajoute la description spécifiée au point de terminaison Verified Access spécifié.

```
aws ec2 modify-verified-access-endpoint \
    --verified-access-endpoint-id vae-066fac616d4d546f2 \
    --description 'Testing Verified Access'
```

Sortie :

```
{
    "VerifiedAccessEndpoint": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
```

```
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
        "ApplicationDomain": "example.com",
        "EndpointType": "network-interface",
        "AttachmentType": "vpc",
        "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
        "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
        "SecurityGroupIds": [
            "sg-004915970c4c8f13a"
        ],
        "NetworkInterfaceOptions": {
            "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
            "Protocol": "https",
            "Port": 443
        },
        "Status": {
            "Code": "updating"
        },
        "Description": "Testing Verified Access",
        "CreationTime": "2023-08-25T20:54:43",
        "LastUpdatedTime": "2023-08-25T22:46:32"
    }
}
```

Pour plus d'informations, consultez la section Points de terminaison Verified Access dans le Guide de l'utilisateur AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessEndpointà la section Référence des AWS CLI commandes.

`modify-verified-access-group-policy`

L'exemple de code suivant montre comment utiliser`modify-verified-access-group-policy`.

AWS CLI

Pour configurer une politique d'accès vérifié pour un groupe

L'`modify-verified-access-group-policy`exemple suivant ajoute la politique d'accès vérifié spécifiée au groupe d'accès vérifié spécifié.

```
aws ec2 modify-verified-access-group-policy \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --policy-enabled \
    --policy-document file://policy.txt
```

Contenu de `policy.txt` :

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

Sortie :

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

Pour plus d'informations, consultez la section Groupes d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessGroupPolicyà la section Référence des AWS CLI commandes.

`modify-verified-access-group`

L'exemple de code suivant montre comment utiliser`modify-verified-access-group`.

AWS CLI

Pour modifier la configuration d'un groupe d'accès vérifié

L'`modify-verified-access-group`exemple suivant ajoute la description spécifiée au groupe d'accès vérifié spécifié.

```
aws ec2 modify-verified-access-group \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
```

```
    --description "Testing Verified Access"
```

Sortie :

```
{
    "VerifiedAccessGroup": {
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "Owner": "123456789012",
        "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
        "CreationTime": "2023-08-25T19:55:19",
        "LastUpdatedTime": "2023-08-25T22:17:25"
    }
}
```

Pour plus d'informations, consultez la section Groupes d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessGroupà la section Référence des AWS CLI commandes.

`modify-verified-access-instance-logging-configuration`

L'exemple de code suivant montre comment utiliser`modify-verified-access-instance-logging-configuration`.

AWS CLI

Pour activer la journalisation pour une instance d'accès vérifié

L'`modify-verified-access-instance-logging-configuration`exemple suivant active la journalisation des accès pour l'instance Verified Access spécifiée. Les journaux seront envoyés au groupe de CloudWatch journaux de journaux spécifié.

```
aws ec2 modify-verified-access-instance-logging-configuration \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}
```

Sortie :

```
{
    "LoggingConfiguration": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "AccessLogs": {
            "S3": {
                "Enabled": false
            },
            "CloudWatchLogs": {
                "Enabled": true,
                "DeliveryStatus": {
                    "Code": "success"
                },
                "LogGroup": "my-log-group"
            },
            "KinesisDataFirehose": {
                "Enabled": false
            },
            "LogVersion": "ocsf-1.0.0-rc.2",
            "IncludeTrustContext": false
        }
    }
}
```

Pour plus d'informations, consultez la section Journaux d'accès vérifiés dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessInstanceLoggingConfigurationà la section Référence des AWS CLI commandes.

`modify-verified-access-instance`

L'exemple de code suivant montre comment utiliser`modify-verified-access-instance`.

AWS CLI

Pour modifier la configuration d'une instance d'accès vérifié

L'`modify-verified-access-instance`exemple suivant ajoute la description spécifiée à l'instance d'accès vérifié spécifiée.

```
aws ec2 modify-verified-access-instance \
```

```
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --description "Testing Verified Access"
```

Sortie :

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [
            {
                "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
                "TrustProviderType": "user",
                "UserTrustProviderType": "iam-identity-center"
            }
        ],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T22:41:04"
    }
}
```

Pour plus d'informations, consultez la section Instances d'accès vérifié dans le guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessInstanceà la section Référence des AWS CLI commandes.

`modify-verified-access-trust-provider`

L'exemple de code suivant montre comment utiliser`modify-verified-access-trust-provider`.

AWS CLI

Pour modifier la configuration d'un fournisseur de confiance Verified Access

L'`modify-verified-access-trust-provider`exemple suivant ajoute la description spécifiée au fournisseur de confiance Verified Access spécifié.

```
aws ec2 modify-verified-access-trust-provider \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
    --description "Testing Verified Access"
```

Sortie :

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "Testing Verified Access",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T19:00:38",
        "LastUpdatedTime": "2023-08-25T19:18:21"
    }
}
```

Pour plus d'informations, consultez la section Trust providers for Verified Access dans le Guide de l'utilisateur de AWS Verified Access.

- Pour plus de détails sur l'API, reportez-vous ModifyVerifiedAccessTrustProviderà la section Référence des AWS CLI commandes.

`modify-volume-attribute`

L'exemple de code suivant montre comment utiliser`modify-volume-attribute`.

AWS CLI

Pour modifier un attribut de volume

Cet exemple définit l'`autoEnableIo`attribut du volume avec l'ID `vol-1234567890abcdef0` sur`true`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- Pour plus de détails sur l'API, reportez-vous ModifyVolumeAttributeà la section Référence des AWS CLI commandes.

`modify-volume`

L'exemple de code suivant montre comment utiliser`modify-volume`.

AWS CLI

Exemple 1 : pour modifier un volume en modifiant sa taille

L'`modify-volume`exemple suivant modifie la taille du volume spécifié à 150 Go.

Commande :

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

Sortie :

```
{
    "VolumeModification": {
        "TargetSize": 150,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": " vol-1234567890abcdef0",
        "TargetIops": 100,
        "StartTime": "2019-05-17T11:27:19.000Z",
        "Progress": 0,
        "OriginalVolumeType": "io1",
        "OriginalIops": 100,
        "OriginalSize": 100
    }
}
```

Exemple 2 : pour modifier un volume en modifiant son type, sa taille et sa valeur d'IOPS

L'`modify-volume`exemple suivant change le type de volume en SSD IOPS provisionné, définit le taux d'IOPS cible à 10000 et définit la taille du volume à 350 Go.

```
aws ec2 modify-volume \
    --volume-type io1 \
    --iops 10000 \
    --size 350 \
    --volume-id vol-1234567890abcdef0
```

Sortie :

```
{
```

```
    "VolumeModification": {
        "TargetSize": 350,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": "vol-0721c1a9d08c93bf6",
        "TargetIops": 10000,
        "StartTime": "2019-05-17T11:38:57.000Z",
        "Progress": 0,
        "OriginalVolumeType": "gp2",
        "OriginalIops": 150,
        "OriginalSize": 50
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVolumeà la section Référence des AWS CLI commandes.

`modify-vpc-attribute`

L'exemple de code suivant montre comment utiliser`modify-vpc-attribute`.

AWS CLI

Pour modifier l' enableDnsSupport attribut

Cet exemple modifie l'`enableDnsSupport`attribut. Cet attribut indique si la résolution DNS est activée pour le VPC. Si cet attribut est `true`, le serveur Amazon DNS résout les noms d'hôte DNS de vos instances par rapport aux adresses IP correspondantes. Dans le cas contraire, il ne le fait pas. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value
\":false}"
```

Pour modifier l' enableDnsHostnames attribut

Cet exemple modifie l'`enableDnsHostnames`attribut. Cet attribut indique si les instances lancées dans le VPC obtiennent des noms d'hôte DNS. Si l'attribut est défini sur `true`, les instances incluses dans le VPC ont obtenu des noms d'hôte DNS. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value
\":false}"
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcAttributeà la section Référence des AWS CLI commandes.

`modify-vpc-endpoint-connection-notification`

L'exemple de code suivant montre comment utiliser`modify-vpc-endpoint-connection-notification`.

AWS CLI

Pour modifier une notification de connexion d'un point de terminaison

Cet exemple modifie la rubrique SNS pour la notification de connexion au point de terminaison spécifiée.

Commande :

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-
id vpce-nfn-008776de7e03f5abc --connection-events Accept Reject --connection-
notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

Sortie :

```
{
   "ReturnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcEndpointConnectionNotificationà la section Référence des AWS CLI commandes.

`modify-vpc-endpoint-service-configuration`

L'exemple de code suivant montre comment utiliser`modify-vpc-endpoint-service-configuration`.

AWS CLI

Pour modifier la configuration d'un service de point de terminaison

Cet exemple modifie l'exigence d'acceptation pour le service de point de terminaison spécifié.

Commande :

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-
svc-09222513e6e77dc86 --no-acceptance-required
```

Sortie :

```
{
   "ReturnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcEndpointServiceConfigurationà la section Référence des AWS CLI commandes.

`modify-vpc-endpoint-service-payer-responsibility`

L'exemple de code suivant montre comment utiliser`modify-vpc-endpoint-service-payer-responsibility`.

AWS CLI

Pour modifier la responsabilité du payeur

L'`modify-vpc-endpoint-service-payer-responsibility`exemple suivant modifie la responsabilité du payeur du service de point de terminaison spécifié.

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \
    --service-id vpce-svc-071afff70666e61e0 \
    --payer-responsibility ServiceOwner
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous ModifyVpcEndpointServicePayerResponsibilityà la section Référence des AWS CLI commandes.

`modify-vpc-endpoint-service-permissions`

L'exemple de code suivant montre comment utiliser`modify-vpc-endpoint-service-permissions`.

AWS CLI

Pour modifier les autorisations des services de point de terminaison

Cet exemple ajoute l'autorisation permettant à un AWS compte de se connecter au service de point de terminaison spécifié.

Commande :

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

Sortie :

```
{
   "ReturnValue": true
}
```

Cet exemple ajoute l'autorisation à un utilisateur IAM spécifique (`admin`) de se connecter au service de point de terminaison spécifié.

Commande :

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/admin"]'
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcEndpointServicePermissionsà la section Référence des AWS CLI commandes.

`modify-vpc-endpoint`

L'exemple de code suivant montre comment utiliser`modify-vpc-endpoint`.

AWS CLI

Pour modifier un point de terminaison de passerelle

Cet exemple modifie le point de terminaison de la passerelle `vpce-1a2b3c4d` en associant la table `rtb-aaa222bb` de routage au point de terminaison et en réinitialisant le document de politique.

Commande :

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-
ids rtb-aaa222bb --reset-policy
```

Sortie :

```
{
  "Return": true
}
```

Pour modifier un point de terminaison d'interface

Cet exemple modifie le point de terminaison de l'interface `vpce-0fe5b17a0707d6fa5` en ajoutant un sous-réseau `subnet-d6fcaa8d` au point de terminaison.

Commande :

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-
id subnet-d6fcaa8d
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcEndpointà la section Référence des AWS CLI commandes.

`modify-vpc-peering-connection-options`

L'exemple de code suivant montre comment utiliser`modify-vpc-peering-connection-options`.

AWS CLI

Pour activer la communication via une connexion d'appairage VPC à partir de votre connexion locale ClassicLink

Dans cet exemple, pour une connexion d'appairage`pcx-aaaabbb`, le propriétaire du VPC demandeur modifie les options de connexion d'appairage du VPC pour permettre à une connexion ClassicLink locale de communiquer avec le VPC homologue.

Commande :

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-
connection-id pcx-aaaabbbb --requester-peering-connection-
options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

Sortie :

```
{
  "RequesterPeeringConnectionOptions": {
      "AllowEgressFromLocalClassicLinkToRemoteVpc": true
  }
}
```

Pour activer la communication via une connexion d'appairage VPC entre votre VPC local et une connexion distante ClassicLink

Dans cet exemple, le propriétaire du VPC accepteur modifie les options de connexion d'appairage du VPC pour permettre au VPC local de communiquer avec la connexion dans le VPC homologue. ClassicLink

Commande :

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-
connection-id pcx-aaaabbbb --accepter-peering-connection-
options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

Sortie :

```
{
  "AccepterPeeringConnectionOptions": {
    "AllowEgressFromLocalVpcToRemoteClassicLink": true
  }
```

```
}
```

Pour activer la prise en charge de la résolution DNS pour la connexion d'appairage VPC

Dans cet exemple, le propriétaire du VPC demandeur modifie les options de connexion d'appairage du VPC afin de permettre au VPC local de résoudre `pcx-aaaabbbb` les noms d'hôte DNS publics en adresses IP privées lorsqu'il est interrogé à partir d'instances du VPC homologue.

Commande :

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

Sortie :

```
{
  "RequesterPeeringConnectionOptions": {
      "AllowDnsResolutionFromRemoteVpc": true
  }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcPeeringConnectionOptionsà la section Référence des AWS CLI commandes.

`modify-vpc-tenancy`

L'exemple de code suivant montre comment utiliser`modify-vpc-tenancy`.

AWS CLI

Pour modifier la location d'un VPC

Cet exemple modifie la location du VPC à. `vpc-1a2b3c4d` `default`

Commande :

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

Sortie :

```
{
```

```
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpcTenancyà la section Référence des AWS CLI commandes.

`modify-vpn-connection-options`

L'exemple de code suivant montre comment utiliser`modify-vpn-connection-options`.

AWS CLI

Pour modifier vos options de connexion VPN

L'`modify-vpn-connection-options`exemple suivant modifie le IPv4 CIDR local du côté de la passerelle client de la connexion VPN spécifiée.

```
aws ec2 modify-vpn-connection-options \
    --vpn-connection-id vpn-1122334455aabbccd \
    --local-ipv4-network-cidr 10.0.0.0/16
```

Sortie :

```
{
    "VpnConnections": [
        {
            "CustomerGatewayConfiguration": "...configuration information...",
            "CustomerGatewayId": "cgw-01234567abcde1234",
            "Category": "VPN",
            "State": "modifying",
            "Type": "ipsec.1",
            "VpnConnectionId": "vpn-1122334455aabbccd",
            "TransitGatewayId": "tgw-00112233445566aab",
            "Options": {
                "EnableAcceleration": false,
                "StaticRoutesOnly": true,
                "LocalIpv4NetworkCidr": "10.0.0.0/16",
                "RemoteIpv4NetworkCidr": "0.0.0.0/0",
                "TunnelInsideIpVersion": "ipv4"
            },
            "Routes": [],
            "Tags": [
```

```
                {
                    "Key": "Name",
                    "Value": "CanadaVPN"
                }
            ],
            "VgwTelemetry": [
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-07-29T10:35:11.000Z",
                    "OutsideIpAddress": "203.0.113.3",
                    "Status": "DOWN",
                    "StatusMessage": ""
                },
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-09-02T09:09:33.000Z",
                    "OutsideIpAddress": "203.0.113.5",
                    "Status": "UP",
                    "StatusMessage": ""
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Modification des options de connexion Site-to-Site VPN dans le Guide de l'utilisateur du AWS Site-to-Site VPN.

- Pour plus de détails sur l'API, reportez-vous ModifyVpnConnectionOptionsà la section Référence des AWS CLI commandes.

`modify-vpn-connection`

L'exemple de code suivant montre comment utiliser`modify-vpn-connection`.

AWS CLI

Pour modifier une connexion VPN

L'`modify-vpn-connection`exemple suivant change la passerelle cible pour la connexion VPN `vpn-12345678901234567` en passerelle privée virtuelle `vgw-11223344556677889` :

```
aws ec2 modify-vpn-connection \
```

```
    --vpn-connection-id vpn-12345678901234567 \
    --vpn-gateway-id vgw-11223344556677889
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "Category": "VPN",
        "State": "modifying",
        "Type": "ipsec.1",
        "VpnConnectionId": "vpn-12345678901234567",
        "VpnGatewayId": "vgw-11223344556677889",
        "Options": {
            "StaticRoutesOnly": false
        },
        "VgwTelemetry": [
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-07-17T07:34:00.000Z",
                "OutsideIpAddress": "18.210.3.222",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            },
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-07-20T21:20:16.000Z",
                "OutsideIpAddress": "34.193.129.33",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpnConnectionà la section Référence des AWS CLI commandes.

`modify-vpn-tunnel-certificate`

L'exemple de code suivant montre comment utiliser`modify-vpn-tunnel-certificate`.

AWS CLI

Pour faire pivoter un certificat de tunnel VPN

L'`modify-vpn-tunnel-certificate`exemple suivant fait pivoter le certificat du tunnel spécifié pour une connexion VPN

```
aws ec2 modify-vpn-tunnel-certificate \
    --vpn-tunnel-outside-ip-address 203.0.113.17 \
    --vpn-connection-id vpn-12345678901234567
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": ...configuration information...,
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "Category": "VPN",
        "State": "modifying",
        "Type": "ipsec.1",
        "VpnConnectionId": "vpn-12345678901234567",
        "VpnGatewayId": "vgw-11223344556677889",
        "Options": {
            "StaticRoutesOnly": false
        },
        "VgwTelemetry": [
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-11T17:27:14.000Z",
                "OutsideIpAddress": "203.0.113.17",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN",
                "CertificateArn": "arn:aws:acm:us-east-1:123456789101:certificate/
c544d8ce-20b8-4fff-98b0-example"
            },
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-11T17:26:47.000Z",
                "OutsideIpAddress": "203.0.114.18",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN",
                "CertificateArn": "arn:aws:acm:us-
east-1:123456789101:certificate/5ab64566-761b-4ad3-b259-example"
```

```
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpnTunnelCertificateà la section Référence des AWS CLI commandes.

`modify-vpn-tunnel-options`

L'exemple de code suivant montre comment utiliser`modify-vpn-tunnel-options`.

AWS CLI

Pour modifier les options du tunnel pour une connexion VPN

L'`modify-vpn-tunnel-options`exemple suivant met à jour les groupes Diffie-Hellman autorisés pour le tunnel et la connexion VPN spécifiés.

```
aws ec2 modify-vpn-tunnel-options \
    --vpn-connection-id vpn-12345678901234567 \
    --vpn-tunnel-outside-ip-address 203.0.113.17 \
    --tunnel-options Phase1DHGroupNumbers=[{Value=14},{Value=15},{Value=16},
{Value=17},{Value=18}],Phase2DHGroupNumbers=[{Value=14},{Value=15},{Value=16},
{Value=17},{Value=18}]
```

Sortie :

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "Category": "VPN",
        "State": "available",
        "Type": "ipsec.1",
        "VpnConnectionId": "vpn-12345678901234567",
        "VpnGatewayId": "vgw-11223344556677889",
        "Options": {
            "StaticRoutesOnly": false,
            "TunnelOptions": [
                {
                    "OutsideIpAddress": "203.0.113.17",
```

```
                    "Phase1DHGroupNumbers": [
                        {
                            "Value": 14
                        },
                        {
                            "Value": 15
                        },
                        {
                            "Value": 16
                        },
                        {
                            "Value": 17
                        },
                        {
                            "Value": 18
                        }
                    ],
                    "Phase2DHGroupNumbers": [
                        {
                            "Value": 14
                        },
                        {
                            "Value": 15
                        },
                        {
                            "Value": 16
                        },
                        {
                            "Value": 17
                        },
                        {
                            "Value": 18
                        }
                    ]
                },
                {
                    "OutsideIpAddress": "203.0.114.19"
                }
            ]
        },
        "VgwTelemetry": [
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-10T21:56:54.000Z",
```

```
                "OutsideIpAddress": "203.0.113.17",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            },
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-10T21:56:43.000Z",
                "OutsideIpAddress": "203.0.114.19",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyVpnTunnelOptionsà la section Référence des AWS CLI commandes.

`monitor-instances`

L'exemple de code suivant montre comment utiliser`monitor-instances`.

AWS CLI

Activation de la surveillance détaillée d'une instance

Cet exemple de commande active la surveillance détaillée de l'instance spécifiée.

Commande :

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
  "InstanceMonitorings": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "Monitoring": {
              "State": "pending"
          }
      }
  ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous MonitorInstancesà la section Référence des AWS CLI commandes.

`move-address-to-vpc`

L'exemple de code suivant montre comment utiliser`move-address-to-vpc`.

AWS CLI

Pour déplacer une adresse vers EC2 -VPC

Cet exemple déplace l'adresse IP élastique 54.123.4.56 vers la plateforme -VPC. EC2

Commande :

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

Sortie :

```
{
  "Status": "MoveInProgress"
}
```

- Pour plus de détails sur l'API, reportez-vous MoveAddressToVpcà la section Référence des AWS CLI commandes.

`move-byoip-cidr-to-ipam`

L'exemple de code suivant montre comment utiliser`move-byoip-cidr-to-ipam`.

AWS CLI

Pour transférer un CIDR BYOIP vers IPAM

L'`move-byoip-cidr-to-ipam`exemple suivant transfère un CIDR BYOIP vers IPAM.

(Linux) :

```
aws ec2 move-byoip-cidr-to-ipam \
    --region us-west-2 \
```

```
    --ipam-pool-id ipam-pool-0a03d430ca3f5c035 \
    --ipam-pool-owner 111111111111 \
    --cidr 130.137.249.0/24
```

(Fenêtres) :

```
aws ec2 move-byoip-cidr-to-ipam ^
    --region us-west-2 ^
    --ipam-pool-id ipam-pool-0a03d430ca3f5c035 ^
    --ipam-pool-owner 111111111111 ^
    --cidr 130.137.249.0/24
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "130.137.249.0/24",
        "State": "pending-transfer"
    }
}
```

Pour plus d'informations, consultez Tutoriel : Transférer un IPv4 CIDR BYOIP existant vers IPAM dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous MoveByoipCidrToIpamà la section Référence des AWS CLI commandes.

`network-insights-access-scope`

L'exemple de code suivant montre comment utiliser`network-insights-access-scope`.

AWS CLI

Pour créer des étendues d'accès à Network Insights

L'`create-network-insights-access-scope`exemple suivant crée une étendue d'accès à Network Insights dans votre AWS compte.

```
aws ec2 create-network-insights-access-scope \
    --cli-input-json file://access-scope-file.json
```

Contenu de `access-scope-file.json` :

```
{
    {
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                         "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

Sortie :

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111"
    }{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-123456789222",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789222:network-insights-access-scope/nis-123456789222",
        "CreatedDate": "2022-01-25T19:20:28.796000+00:00",
        "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
    },
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-04c0c0fbca737c404",
        "MatchPaths": [
            {
```

```
                "Source": {
                    "ResourceStatement": {
                        "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous NetworkInsightsAccessScopeà la section Référence des AWS CLI commandes.

`provision-byoip-cidr`

L'exemple de code suivant montre comment utiliser`provision-byoip-cidr`.

AWS CLI

Pour provisionner une plage d'adresses

L'`provision-byoip-cidr`exemple suivant fournit une plage d'adresses IP publiques à utiliser avec AWS.

```
aws ec2 provision-byoip-cidr \
    --cidr 203.0.113.25/24 \
```

```
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-provision"
    }
}
```

Pour plus d'informations sur la création des chaînes de messages pour le contexte d'autorisation, consultez Bring Your Own IP Addresses dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ProvisionByoipCidrà la section Référence des AWS CLI commandes.

`provision-ipam-pool-cidr`

L'exemple de code suivant montre comment utiliser`provision-ipam-pool-cidr`.

AWS CLI

Pour fournir un CIDR à un pool IPAM

L'`provision-ipam-pool-cidr`exemple suivant fournit un CIDR à un pool IPAM.

(Linux) :

```
aws ec2 provision-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --cidr 10.0.0.0/24
```

(Fenêtres) :

```
aws ec2 provision-ipam-pool-cidr ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --cidr 10.0.0.0/24
```

Sortie :

```
{
```

```
    "IpamPoolCidr": {
        "Cidr": "10.0.0.0/24",
        "State": "pending-provision"
    }
}
```

Pour plus d'informations, consultez la section CIDRs Provisionner un pool dans le guide de l'utilisateur Amazon VPC IPAM.

- Pour plus de détails sur l'API, reportez-vous ProvisionIpamPoolCidrà la section Référence des AWS CLI commandes.

`purchase-host-reservation`

L'exemple de code suivant montre comment utiliser`purchase-host-reservation`.

AWS CLI

Pour acheter une réservation d'hôte dédié

Cet exemple achète l'offre de réservation d'hôte dédié spécifiée pour l'hôte dédié spécifié sur votre compte.

Commande :

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

Sortie :

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
      {
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "HostReservationId": "hr-0d418a3a4ffc669ae",
          "UpfrontPrice": "0.000",
```

```
            "Duration": 31536000
        }
    ],
    "TotalUpfrontPrice": "0.000"
}
```

- Pour plus de détails sur l'API, reportez-vous PurchaseHostReservationà la section Référence des AWS CLI commandes.

`purchase-reserved-instances-offering`

L'exemple de code suivant montre comment utiliser`purchase-reserved-instances-offering`.

AWS CLI

Pour acheter une offre d'instance réservée

Cet exemple de commande illustre l'achat d'une offre d'instances réservées, en spécifiant un ID d'offre et un nombre d'instances.

Commande :

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-
id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

Sortie :

```
{
    "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
}
```

- Pour plus de détails sur l'API, reportez-vous PurchaseReservedInstancesOfferingà la section Référence des AWS CLI commandes.

`purchase-scheduled-instances`

L'exemple de code suivant montre comment utiliser`purchase-scheduled-instances`.

AWS CLI

Pour acheter une instance planifiée

Cet exemple achète une instance planifiée.

Commande :

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

Purchase-Request.json :

```
[
    {
        "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
        "InstanceCount": 1
    }
]
```

Sortie :

```
{
  "ScheduledInstanceSet": [
      {
          "AvailabilityZone": "us-west-2b",
          "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
          "HourlyPrice": "0.095",
          "CreateDate": "2016-01-25T21:43:38.612Z",
          "Recurrence": {
              "OccurrenceDaySet": [
                  1
              ],
              "Interval": 1,
              "Frequency": "Weekly",
              "OccurrenceRelativeToEnd": false,
              "OccurrenceUnit": ""
          },
          "Platform": "Linux/UNIX",
          "TermEndDate": "2017-01-31T09:00:00Z",
          "InstanceCount": 1,
          "SlotDurationInHours": 32,
          "TermStartDate": "2016-01-31T09:00:00Z",
          "NetworkPlatform": "EC2-VPC",
          "TotalScheduledInstanceHours": 1696,
          "NextSlotStartTime": "2016-01-31T09:00:00Z",
          "InstanceType": "c4.large"
```

```
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous PurchaseScheduledInstancesà la section Référence des AWS CLI commandes.

`reboot-instances`

L'exemple de code suivant montre comment utiliser`reboot-instances`.

AWS CLI

Pour redémarrer une EC2 instance Amazon

Cet exemple redémarre l'instance spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

Pour plus d'informations, veuillez consulter la rubrique Redémarrer votre instance dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous RebootInstancesà la section Référence des AWS CLI commandes.

`register-image`

L'exemple de code suivant montre comment utiliser`register-image`.

AWS CLI

Exemple 1 : pour enregistrer une AMI à l'aide d'un fichier manifeste

L'`register-image`exemple suivant enregistre une AMI à l'aide du fichier manifeste spécifié dans Amazon S3.

```
aws ec2 register-image \
    --name my-image \
```

```
    --image-location amzn-s3-demo-bucket/myimage/image.manifest.xml
```

Sortie :

```
{
    "ImageId": "ami-1234567890EXAMPLE"
}
```

Pour plus d'informations, consultez Amazon Machine Images (AMI) dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour enregistrer une AMI à l'aide d'un instantané d'un périphérique racine

L'`register-image`exemple suivant enregistre une AMI en utilisant l'instantané spécifié d'un volume racine EBS en tant que périphérique`/dev/xvda`. Le mappage des périphériques en mode bloc inclut également un volume EBS vide de 100 GiB en tant que périphérique. `/dev/xvdf`

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f} DeviceName=/dev/
xvdf,Ebs={VolumeSize=100}
```

Sortie :

```
{
    "ImageId": "ami-1a2b3c4d5eEXAMPLE"
}
```

Pour plus d'informations, consultez Amazon Machine Images (AMI) dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous RegisterImageà la section Référence des AWS CLI commandes.

`register-instance-event-notification-attributes`

L'exemple de code suivant montre comment utiliser`register-instance-event-notification-attributes`.

AWS CLI

Exemple 1 : pour inclure tous les tags dans les notifications d'événements

L'`register-instance-event-notification-attributes`exemple suivant inclut toutes les balises des notifications d'événements.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

Sortie :

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

Pour plus d'informations, consultez la section Événements planifiés pour vos instances dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour inclure des balises spécifiques dans les notifications d'événements

L'`register-instance-event-notification-attributes`exemple suivant inclut les balises spécifiées dans les notifications d'événements. Vous ne pouvez pas spécifier de balises si tel `IncludeAllTagsOfInstance` est le cas`true`.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

Sortie :

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [
            "tag-key1",
            "tag-key2"
        ],
        "IncludeAllTagsOfInstance": false
    }
```

```
}
```

Pour plus d'informations, consultez la section Événements planifiés pour vos instances dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous RegisterInstanceEventNotificationAttributesà la section Référence des AWS CLI commandes.

`register-transit-gateway-multicase-group-sources`

L'exemple de code suivant montre comment utiliser`register-transit-gateway-multicase-group-sources`.

AWS CLI

Pour enregistrer une source auprès d'un groupe de multidiffusion de passerelle de transit.

L'`register-transit-gateway-multicast-group-sources`exemple suivant enregistre la source du groupe d'interfaces réseau spécifiée auprès d'un groupe de multidiffusion.

```
aws ec2 register-transit-gateway-multicast-group-sources \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-07f290fc3c090cbae
```

Sortie :

```
{
    "RegisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

Pour plus d'informations, voir Enregistrer des sources auprès d'un groupe de multidiffusion dans le guide de l'utilisateur de AWS Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous RegisterTransitGatewayMulticaseGroupSourcesà la section Référence des AWS CLI commandes.

`register-transit-gateway-multicast-group-members`

L'exemple de code suivant montre comment utiliser`register-transit-gateway-multicast-group-members`.

AWS CLI

Pour afficher les informations relatives aux associations de domaines de multidiffusion de la passerelle de transit

L'`register-transit-gateway-multicast-group-members`exemple suivant renvoie les associations pour le domaine de multidiffusion spécifié.

```
aws ec2 register-transit-gateway-multicast-group-members \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-0e246d32695012e81
```

Sortie :

```
{
    "RegisteredMulticastGroupMembers": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-0e246d32695012e81"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

Pour plus d'informations, consultez la section Domaines de multidiffusion dans le guide de l'utilisateur de Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous RegisterTransitGatewayMulticastGroupMembersà la section Référence des AWS CLI commandes.

`register-transit-gateway-multicast-group-sources`

L'exemple de code suivant montre comment utiliser`register-transit-gateway-multicast-group-sources`.

AWS CLI

Pour enregistrer une source auprès d'un groupe de multidiffusion de passerelle de transit.

L'`register-transit-gateway-multicast-group-sources`exemple suivant enregistre la source du groupe d'interfaces réseau spécifiée auprès d'un groupe de multidiffusion.

```
aws ec2 register-transit-gateway-multicast-group-sources \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-07f290fc3c090cbae
```

Sortie :

```
{
    "RegisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

Pour plus d'informations, consultez la section Domaines de multidiffusion dans le Guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous RegisterTransitGatewayMulticastGroupSourcesà la section Référence des AWS CLI commandes.

`reject-transit-gateway-peering-attachment`

L'exemple de code suivant montre comment utiliser`reject-transit-gateway-peering-attachment`.

AWS CLI

Pour rejeter une pièce jointe d'appairage d'une passerelle de transit

L'`reject-transit-gateway-peering-attachment`exemple suivant rejette la demande de pièce jointe d'appairage de passerelle de transit spécifiée. Le `--region` paramètre spécifie la région dans laquelle se trouve la passerelle de transit acceptrice.

```
aws ec2 reject-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
    --region us-east-2
```

Sortie :

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "rejecting",
        "CreationTime": "2019-12-09T11:50:31.000Z"
    }
}
```

Pour plus d'informations, consultez la section Transit Gateway peering Attachments dans le guide Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous RejectTransitGatewayPeeringAttachmentà la section Référence des AWS CLI commandes.

`reject-transit-gateway-vpc-attachment`

L'exemple de code suivant montre comment utiliser`reject-transit-gateway-vpc-attachment`.

AWS CLI

Pour rejeter une pièce jointe VPC d'une passerelle de transit

L'`reject-transit-gateway-vpc-attachment`exemple suivant rejette l'attachement VPC de passerelle de transit spécifié.

```
aws ec2 reject-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Sortie :

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

Pour plus d'informations, consultez la section Pièces jointes d'une passerelle de transit à un VPC dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous RejectTransitGatewayVpcAttachmentà la section Référence des AWS CLI commandes.

`reject-transit-gateway-vpc-attachments`

L'exemple de code suivant montre comment utiliser`reject-transit-gateway-vpc-attachments`.

AWS CLI

Pour rejeter une pièce jointe VPC d'une passerelle de transit

L'`reject-transit-gateway-vpc-attachment`exemple suivant rejette l'attachement VPC de passerelle de transit spécifié.

```
aws ec2 reject-transit-gateway-vpc-attachment \
```

```
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Sortie :

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

Pour plus d'informations, consultez la section Pièces jointes d'une passerelle de transit à un VPC dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous RejectTransitGatewayVpcAttachmentsà la section Référence des AWS CLI commandes.

`reject-vpc-endpoint-connections`

L'exemple de code suivant montre comment utiliser`reject-vpc-endpoint-connections`.

AWS CLI

Pour rejeter une demande de connexion à un point de terminaison d'interface

Cet exemple rejette la demande de connexion au point de terminaison spécifié pour le service de point de terminaison spécifié.

Commande :

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Sortie :

```
{
  "Unsuccessful": []
}
```

- Pour plus de détails sur l'API, reportez-vous RejectVpcEndpointConnectionsà la section Référence des AWS CLI commandes.

`reject-vpc-peering-connection`

L'exemple de code suivant montre comment utiliser`reject-vpc-peering-connection`.

AWS CLI

Pour rejeter une connexion d'appairage VPC

Cet exemple rejette la demande de connexion d'appairage VPC spécifiée.

Commande :

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Sortie :

```
{
    "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous RejectVpcPeeringConnectionà la section Référence des AWS CLI commandes.

`release-address`

L'exemple de code suivant montre comment utiliser`release-address`.

AWS CLI

Pour publier une adresse IP élastique pour EC2 -Classic

Cet exemple libère une adresse IP élastique à utiliser avec des instances dans EC2 -Classic. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 release-address --public-ip 198.51.100.0
```

Pour libérer une adresse IP élastique pour EC2 -VPC

Cet exemple libère une adresse IP Elastic pour pouvoir l'utiliser avec des instances dans un VPC. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- Pour plus de détails sur l'API, reportez-vous ReleaseAddressà la section Référence des AWS CLI commandes.

`release-hosts`

L'exemple de code suivant montre comment utiliser`release-hosts`.

AWS CLI

Pour libérer un hôte dédié de votre compte

Pour libérer un hôte dédié de votre compte. Les instances présentes sur l'hôte doivent être arrêtées ou résiliées avant que l'hôte puisse être libéré.

Commande :

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

Sortie :

```
{
    "Successful":  [
        "h-0029d6e3cacf1b3da"
         ],
  "Unsuccessful": []

 }
```

- Pour plus de détails sur l'API, reportez-vous ReleaseHostsà la section Référence des AWS CLI commandes.

`release-ipam-pool-allocation`

L'exemple de code suivant montre comment utiliser`release-ipam-pool-allocation`.

AWS CLI

Pour libérer une allocation de pool IPAM

Dans cet exemple, vous êtes un administrateur délégué IPAM qui a essayé de supprimer un pool IPAM mais qui a reçu un message d'erreur indiquant que vous ne pouvez pas supprimer le pool tant que celui-ci est doté d'allocations. Vous utilisez cette commande pour libérer une allocation de pool.

Remarques :

Vous ne pouvez utiliser cette commande que pour des allocations personnalisées. Pour supprimer une allocation pour une ressource sans supprimer la ressource, définissez son état surveillé sur false à l'aide de modify-ipam-resource-cidr.Pour terminer cette demande, vous aurez besoin de l'ID du pool IPAM, que vous pouvez obtenir avec. describe-ipam-pools Vous aurez également besoin de l'ID d'allocation, que vous pouvez obtenir avec get-ipam-pool-allocations.Si vous ne souhaitez pas supprimer les allocations une par une, vous pouvez utiliser le `--cascade option` lorsque vous supprimez un pool IPAM pour libérer automatiquement toutes les allocations du pool avant de le supprimer.Il existe un certain nombre de conditions préalables avant d'exécuter cette commande. Pour plus d'informations, consultez la section Publier une allocation dans le guide de l'utilisateur Amazon VPC IPAM. L'environnement `--region` dans lequel vous exécutez cette commande doit correspondre aux paramètres régionaux du pool IPAM dans lequel se trouve l'allocation.

L'`release-ipam-pool-allocation`exemple suivant publie une allocation de pool IPAM.

```
aws ec2 release-ipam-pool-allocation \
    --ipam-pool-id ipam-pool-07bdd12d7c94e4693 \
    --cidr 10.0.0.0/23 \
    --ipam-pool-allocation-id ipam-pool-alloc-0e66a1f730da54791b99465b79e7d1e89 \
    --region us-west-1
```

Sortie :

```
{
    "Success": true
}
```

Une fois que vous avez publié une allocation, vous souhaiterez peut-être vous lancer delete-ipam-pool.

- Pour plus de détails sur l'API, reportez-vous ReleaseIpamPoolAllocationà la section Référence des AWS CLI commandes.

`replace-iam-instance-profile-association`

L'exemple de code suivant montre comment utiliser`replace-iam-instance-profile-association`.

AWS CLI

Remplacement d'un profil d'instance IAM pour une instance

Cet exemple remplace le profil d'instance IAM représenté par l'association `iip-assoc-060bae234aac2e7fa` par le profil d'instance IAM nommé `AdminRole`.

```
aws ec2 replace-iam-instance-profile-association \
    --iam-instance-profile Name=AdminRole \
    --association-id iip-assoc-060bae234aac2e7fa
```

Sortie :

```
{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-087711ddaf98f9489",
        "State": "associating",
```

```
        "AssociationId": "iip-assoc-0b215292fab192820",
        "IamInstanceProfile": {
            "Id": "AIPAJLNLDX3AMYZNWYYAY",
            "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ReplaceIamInstanceProfileAssociationà la section Référence des AWS CLI commandes.

`replace-network-acl-association`

L'exemple de code suivant montre comment utiliser`replace-network-acl-association`.

AWS CLI

Pour remplacer l'ACL réseau associée à un sous-réseau

Cet exemple associe l'ACL réseau spécifiée au sous-réseau pour l'association ACL réseau spécifiée.

Commande :

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --
network-acl-id acl-5fb85d36
```

Sortie :

```
{
    "NewAssociationId": "aclassoc-3999875b"
}
```

- Pour plus de détails sur l'API, reportez-vous ReplaceNetworkAclAssociationà la section Référence des AWS CLI commandes.

`replace-network-acl-entry`

L'exemple de code suivant montre comment utiliser`replace-network-acl-entry`.

AWS CLI

Pour remplacer une entrée ACL réseau

Cet exemple remplace une entrée pour l'ACL réseau spécifiée. La nouvelle règle 100 autorise le trafic entrant du 203.0.113.12/24 sur le port UDP 53 (DNS) vers n'importe quel sous-réseau associé.

Commande :

```
aws ec2 replace-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 203.0.113.12/24 --rule-action allow
```

- Pour plus de détails sur l'API, reportez-vous ReplaceNetworkAclEntryà la section Référence des AWS CLI commandes.

`replace-route-table-association`

L'exemple de code suivant montre comment utiliser`replace-route-table-association`.

AWS CLI

Pour remplacer la table de routage associée à un sous-réseau

Cet exemple associe la table de routage spécifiée au sous-réseau pour l'association de table de routage spécifiée.

Commande :

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

Sortie :

```
{
    "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- Pour plus de détails sur l'API, reportez-vous ReplaceRouteTableAssociationà la section Référence des AWS CLI commandes.

`replace-route`

L'exemple de code suivant montre comment utiliser`replace-route`.

AWS CLI

Pour remplacer un itinéraire

Cet exemple remplace l'itinéraire spécifié dans la table de routage spécifiée. La nouvelle route correspond au CIDR spécifié et envoie le trafic vers la passerelle privée virtuelle spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-
block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- Pour plus de détails sur l'API, reportez-vous ReplaceRouteà la section Référence des AWS CLI commandes.

`replace-transit-gateway-route`

L'exemple de code suivant montre comment utiliser`replace-transit-gateway-route`.

AWS CLI

Pour remplacer l'itinéraire spécifié dans la table de routage de la passerelle de transit spécifiée

L'`replace-transit-gateway-route`exemple suivant remplace l'itinéraire dans la table de routage de la passerelle de transit spécifiée.

```
aws ec2 replace-transit-gateway-route \
    --destination-cidr-block 10.0.2.0/24 \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

Sortie :

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
```

```
            {
                "ResourceId": "vpc-4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "active"
    }
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous ReplaceTransitGatewayRouteà la section Référence des AWS CLI commandes.

`report-instance-status`

L'exemple de code suivant montre comment utiliser`report-instance-status`.

AWS CLI

Pour signaler des commentaires sur le statut d'une instance

Cet exemple de commande indique le statut de l'instance spécifiée.

Commande :

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --
reason-codes unresponsive
```

- Pour plus de détails sur l'API, reportez-vous ReportInstanceStatusà la section Référence des AWS CLI commandes.

`request-spot-fleet`

L'exemple de code suivant montre comment utiliser`request-spot-fleet`.

AWS CLI

Pour demander une flotte Spot dans le sous-réseau au prix le plus bas

Cet exemple de commande crée une demande de flotte Spot avec deux spécifications de lancement qui ne diffèrent que selon le sous-réseau. La flotte Spot lance les instances dans le sous-réseau spécifié au prix le plus bas. Si les instances sont lancées dans un VPC par défaut, elles reçoivent une adresse IP publique par défaut. Si les instances sont lancées sur un VPC personnalisé, elles ne reçoivent pas d'adresse IP publique par défaut.

Notez que vous ne pouvez pas spécifier différents sous-réseaux provenant de la même zone de disponibilité dans une demande de flotte Spot.

Commande :

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json :

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "SubnetId": "subnet-1a2b3c4d, subnet-3c4d5e6f",
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

Sortie :

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
```

```
}
```

Pour demander une flotte Spot dans la zone de disponibilité au prix le plus bas

Cet exemple de commande crée une demande de flotte Spot avec deux spécifications de lancement qui ne diffèrent que selon la zone de disponibilité. La flotte Spot lance les instances dans la zone de disponibilité spécifiée au prix le plus bas. Si votre compte prend uniquement en charge l'option EC2 -VPC, Amazon EC2 lance les instances Spot dans le sous-réseau par défaut de la zone de disponibilité. Si votre compte est compatible avec EC2 -Classic, Amazon EC2 lance les instances dans EC2 -Classic dans la zone de disponibilité.

Commande :

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json :

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "Placement": {
              "AvailabilityZone": "us-west-2a, us-west-2b"
          },
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

Pour lancer des instances Spot dans un sous-réseau et leur attribuer des adresses IP publiques

Cet exemple de commande attribue des adresses publiques aux instances lancées dans un VPC autre que celui par défaut. Notez que lorsque vous spécifiez une interface réseau, vous devez inclure l'ID du sous-réseau et l'ID du groupe de sécurité à l'aide de l'interface réseau.

Commande :

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json :

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "InstanceType": "m3.medium",
          "NetworkInterfaces": [
              {
                  "DeviceIndex": 0,
                  "SubnetId": "subnet-1a2b3c4d",
                  "Groups": [ "sg-1a2b3c4d" ],
                  "AssociatePublicIpAddress": true
              }
          ],
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
          }
      }
  ]
}
```

Pour demander une flotte Spot en utilisant la stratégie d'allocation diversifiée

Cet exemple de commande crée une demande de flotte Spot qui lance 30 instances en utilisant la stratégie d'allocation diversifiée. Les spécifications de lancement varient selon le type d'instance. La flotte Spot répartit les instances selon les spécifications de lancement, de sorte qu'il existe 10 instances de chaque type.

Commande :

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json :

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "m3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      }
  ]
}
```

Pour plus d'informations, consultez la section Spot Fleet Requests dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous RequestSpotFleetà la section Référence des AWS CLI commandes.

`request-spot-instances`

L'exemple de code suivant montre comment utiliser`request-spot-instances`.

AWS CLI

Pour demander des instances Spot

Cet exemple de commande crée une demande d'instance ponctuelle unique pour cinq instances dans la zone de disponibilité spécifiée. Si votre compte prend uniquement en charge EC2 -VPC, Amazon EC2 lance les instances dans le sous-réseau par défaut de la zone de disponibilité spécifiée. Si votre compte prend en charge EC2 -Classic, Amazon EC2 lance les instances dans EC2 -Classic dans la zone de disponibilité spécifiée.

Commande :

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Spécification.json :

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Sortie :

```
{
  "SpotInstanceRequests": [
      {
          "Status": {
              "UpdateTime": "2014-03-25T20:54:21.000Z",
              "Code": "pending-evaluation",
              "Message": "Your Spot request has been submitted for review, and is
 pending evaluation."
          },
          "ProductDescription": "Linux/UNIX",
          "SpotInstanceRequestId": "sir-df6f405d",
          "State": "open",
          "LaunchSpecification": {
              "Placement": {
```

```
                "AvailabilityZone": "us-west-2a"
            },
            "ImageId": "ami-1a2b3c4d",
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                {
                    "GroupName": "my-security-group",
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "Monitoring": {
                "Enabled": false
            },
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            },
            "InstanceType": "m3.medium"
        },
        "Type": "one-time",
        "CreateTime": "2014-03-25T20:54:20.000Z",
        "SpotPrice": "0.050000"
    },
    ...
  ]
}
```

Cet exemple de commande crée une demande d'instance Spot unique pour cinq instances dans le sous-réseau spécifié. Amazon EC2 lance les instances dans le sous-réseau spécifié. Si le VPC n'est pas un VPC par défaut, les instances ne reçoivent pas d'adresse IP publique par défaut.

Commande :

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Spécification.json :

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
```

```
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Sortie :

```
{
  "SpotInstanceRequests": [
      {
          "Status": {
             "UpdateTime": "2014-03-25T22:21:58.000Z",
             "Code": "pending-evaluation",
             "Message": "Your Spot request has been submitted for review, and is
 pending evaluation."
          },
          "ProductDescription": "Linux/UNIX",
          "SpotInstanceRequestId": "sir-df6f405d",
          "State": "open",
          "LaunchSpecification": {
             "Placement": {
                 "AvailabilityZone": "us-west-2a"
             }
             "ImageId": "ami-1a2b3c4d"
             "SecurityGroups": [
                 {
                     "GroupName": "my-security-group",
                     "GroupID": "sg-1a2b3c4d"
                 }
             ]
             "SubnetId": "subnet-1a2b3c4d",
             "Monitoring": {
                 "Enabled": false
             },
             "IamInstanceProfile": {
                 "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
             },
             "InstanceType": "m3.medium",
          },
          "Type": "one-time",
          "CreateTime": "2014-03-25T22:21:58.000Z",
          "SpotPrice": "0.050000"
      },
```

```
      ...
  ]
}
```

Cet exemple attribue une adresse IP publique aux instances Spot que vous lancez dans un VPC autre que celui par défaut. Notez que lorsque vous spécifiez une interface réseau, vous devez inclure l'ID du sous-réseau et l'ID du groupe de sécurité à l'aide de l'interface réseau.

Commande :

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

Spécification.json :

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
      "Groups": [ "sg-1a2b3c4d" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous RequestSpotInstancesà la section Référence des AWS CLI commandes.

`reset-address-attribute`

L'exemple de code suivant montre comment utiliser`reset-address-attribute`.

AWS CLI

Pour réinitialiser l'attribut de nom de domaine associé à une adresse IP élastique

Les `reset-address-attribute` exemples suivants réinitialisent l'attribut de nom de domaine d'une adresse IP élastique.

Linux :

```
aws ec2 reset-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --attribute domain-name
```

Windows :

```
aws ec2 reset-address-attribute ^
    --allocation-id eipalloc-abcdef01234567890 ^
    --attribute domain-name
```

Sortie :

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com."
            "PtrRecordUpdate": {
                "Value": "example.net.",
                "Status": "PENDING"
        }
    ]
}
```

Pour surveiller le changement en attente, consultez le manuel describe-addresses-attributede référence des commandes de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous ResetAddressAttributeà la section Référence des AWS CLI commandes.

`reset-ebs-default-kms-key-id`

L'exemple de code suivant montre comment utiliser`reset-ebs-default-kms-key-id`.

AWS CLI

Pour réinitialiser votre clé CMK par défaut pour le chiffrement EBS

L'`reset-ebs-default-kms-key-id`exemple suivant réinitialise la clé CMK par défaut pour le chiffrement EBS de votre AWS compte dans la région actuelle.

```
aws ec2 reset-ebs-default-kms-key-id
```

Sortie :

```
{
   "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513eEXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous ResetEbsDefaultKmsKeyIdà la section Référence des AWS CLI commandes.

`reset-fpga-image-attribute`

L'exemple de code suivant montre comment utiliser`reset-fpga-image-attribute`.

AWS CLI

Pour réinitialiser les attributs d'une image Amazon FPGA

Cet exemple réinitialise les autorisations de chargement pour l'AFI spécifié.

Commande :

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

Sortie :

```
{
  "Return": true
}
```

- Pour plus de détails sur l'API, reportez-vous ResetFpgaImageAttributeà la section Référence des AWS CLI commandes.

`reset-image-attribute`

L'exemple de code suivant montre comment utiliser`reset-image-attribute`.

AWS CLI

Pour réinitialiser l'attribut LaunchPermission

Cet exemple rétablit la valeur par défaut de l'`launchPermission`attribut de l'AMI spécifiée. Par défaut, AMIs sont privés. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute launchPermission
```

- Pour plus de détails sur l'API, reportez-vous ResetImageAttributeà la section Référence des AWS CLI commandes.

`reset-instance-attribute`

L'exemple de code suivant montre comment utiliser`reset-instance-attribute`.

AWS CLI

Pour réinitialiser l' sourceDestCheck attribut

Cet exemple réinitialise l'`sourceDestCheck`attribut de l'instance spécifiée. L'instance doit se trouver dans un VPC. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute sourceDestCheck
```

Pour réinitialiser l'attribut du noyau

Cet exemple réinitialise l'`kernel`attribut de l'instance spécifiée. L'instance doit être dans l'état `stopped`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute kernel
```

Pour réinitialiser l'attribut ramdisk

Cet exemple réinitialise l'`ramdisk`attribut de l'instance spécifiée. L'instance doit être dans l'état `stopped`. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute ramdisk
```

- Pour plus de détails sur l'API, reportez-vous ResetInstanceAttributeà la section Référence des AWS CLI commandes.

`reset-network-interface-attribute`

L'exemple de code suivant montre comment utiliser`reset-network-interface-attribute`.

AWS CLI

Pour réinitialiser un attribut d'interface réseau

L'`reset-network-interface-attribute`exemple suivant réinitialise la valeur de l'attribut de source/destination vérification sur`true`.

```
aws ec2 reset-network-interface-attribute \
    --network-interface-id eni-686ea200 \
    --source-dest-check
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous ResetNetworkInterfaceAttributeà la section Référence des AWS CLI commandes.

`reset-snapshot-attribute`

L'exemple de code suivant montre comment utiliser`reset-snapshot-attribute`.

AWS CLI

Pour réinitialiser un attribut de capture d'écran

Cet exemple réinitialise les autorisations de création de volume pour les instantanéssnap-1234567890abcdef0. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --
attribute createVolumePermission
```

- Pour plus de détails sur l'API, reportez-vous ResetSnapshotAttributeà la section Référence des AWS CLI commandes.

restore-address-to-classic

L'exemple de code suivant montre comment utiliserrestore-address-to-classic.

AWS CLI

Pour restaurer une adresse dans EC2 -Classic

Cet exemple restaure l'adresse IP Elastic 198.51.100.0 sur la plateforme -Classic. EC2

Commande :

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

Sortie :

```
{
  "Status": "MoveInProgress",
  "PublicIp": "198.51.100.0"
}
```

- Pour plus de détails sur l'API, reportez-vous RestoreAddressToClassicà la section Référence des AWS CLI commandes.

`restore-image-from-recycle-bin`

L'exemple de code suivant montre comment utiliser`restore-image-from-recycle-bin`.

AWS CLI

Pour restaurer une image depuis la corbeille

L'`restore-image-from-recycle-bin`exemple suivant restaure l'AMI ami-0111222333444abcd à partir de la corbeille.

```
aws ec2 restore-image-from-recycle-bin \
    --image-id ami-0111222333444abcd
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez la section Récupérer les AMIs fichiers supprimés de la corbeille dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous RestoreImageFromRecycleBinà la section Référence des AWS CLI commandes.

`restore-managed-prefix-list-version`

L'exemple de code suivant montre comment utiliser`restore-managed-prefix-list-version`.

AWS CLI

US-West-2**Pour restaurer la version d'une liste de préfixes**

Ce qui suit `restore-managed-prefix-list-version` restaure les entrées de la version 1 de la liste de préfixes spécifiée.

```
aws ec2 restore-managed-prefix-list-version \
    --prefix-list-id pl-0123456abcabcabc1 \
    --current-version 2 \
    --previous-version 1
```

Sortie :

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "restore-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 2,
        "OwnerId": "123456789012"
    }
}
```

Pour plus d'informations, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous RestoreManagedPrefixListVersionà la section Référence des AWS CLI commandes.

`restore-snapshot-from-recycle-bin`

L'exemple de code suivant montre comment utiliser`restore-snapshot-from-recycle-bin`.

AWS CLI

Pour restaurer des instantanés depuis la corbeille

L'`restore-snapshot-from-recycle-bin`exemple suivant restaure un instantané à partir de la corbeille. Lorsque vous restaurez un instantané à partir de la corbeille, il est immédiatement disponible pour utilisation et il est supprimé de la corbeille. Une fois qu'il est restauré, vous pouvez l'utiliser de la même manière que n'importe quel autre instantané de votre compte.

```
aws ec2 restore-snapshot-from-recycle-bin \
    --snapshot-id snap-01234567890abcdef
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur la corbeille, consultez la section Récupérer des instantanés supprimés de la corbeille dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous RestoreSnapshotFromRecycleBinà la section Référence des AWS CLI commandes.

`restore-snapshot-tier`

L'exemple de code suivant montre comment utiliser`restore-snapshot-tier`.

AWS CLI

Exemple 1 : pour restaurer définitivement un instantané archivé

L'`restore-snapshot-tier`exemple suivant restaure définitivement le cliché spécifié. Spécifiez l'option `--snapshot-id` et incluez l'`permanent-restore`option.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --permanent-restore
```

Sortie :

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

Pour plus d'informations sur l'archivage des instantanés, consultez la section Archiver des instantanés Amazon EBS dans le guide de l'utilisateur Amazon EBS.

Exemple 2 : pour restaurer temporairement un instantané archivé

L'`restore-snapshot-tier`exemple suivant restaure temporairement le cliché spécifié. Omettez l'option `--permanent-restore`. Spécifiez le `--snapshot-id` et, pour`temporary-restore-days`, le nombre de jours pendant lesquels vous souhaitez restaurer le cliché. `temporary-restore-days`doit être spécifiée en jours. La plage autorisée est 1 de`180`. Si vous ne spécifiez aucune valeur, la valeur par défaut est le jour 1.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --temporary-restore-days 5
```

Sortie :

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

Pour plus d'informations sur l'archivage des instantanés, consultez la section Archiver des instantanés Amazon EBS dans le guide de l'utilisateur Amazon EBS.

Exemple 3 : pour modifier la période de restauration

L'`restore-snapshot-tier`exemple suivant modifie la période de restauration pour le cliché spécifié en `10` jours.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --temporary-restore-days 10
```

Sortie :

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 10,
    "IsPermanentRestore": false
}
```

Pour plus d'informations sur l'archivage des instantanés, consultez la section Archiver des instantanés Amazon EBS dans le guide de l'utilisateur Amazon EBS.

Exemple 4 : pour modifier le type de restauration

L'`restore-snapshot-tier`exemple suivant change le type de restauration pour le cliché spécifié de temporaire à permanent.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --permanent-restore
```

Sortie :

```
{
```

```
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

Pour plus d'informations sur l'archivage des instantanés, consultez la section Archiver des instantanés Amazon EBS dans le guide de l'utilisateur Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous RestoreSnapshotTierà la section Référence des AWS CLI commandes.

`revoke-client-vpn-ingress`

L'exemple de code suivant montre comment utiliser`revoke-client-vpn-ingress`.

AWS CLI

Pour révoquer une règle d'autorisation pour un point de terminaison VPN client

L'`revoke-client-vpn-ingress`exemple suivant révoque une règle d'accès à Internet (`0.0.0.0/0`) pour tous les groupes.

```
aws ec2 revoke-client-vpn-ingress \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

Sortie :

```
{
    "Status": {
        "Code": "revoking"
    }
}
```

Pour plus d'informations, consultez la section Règles d'autorisation dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous RevokeClientVpnIngressà la section Référence des AWS CLI commandes.

`revoke-security-group-egress`

L'exemple de code suivant montre comment utiliser`revoke-security-group-egress`.

AWS CLI

Exemple 1 : pour supprimer la règle qui autorise le trafic sortant vers une plage d'adresses spécifique

L'`revoke-security-group-egress`exemple de commande suivant supprime la règle qui autorise l'accès aux plages d'adresses spécifiées sur le port TCP 80.

```
aws ec2 revoke-security-group-egress \
    --group-id sg-026c12253ce15eff7 \
    --ip-
permissions [{IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour supprimer la règle qui autorise le trafic sortant vers un groupe de sécurité spécifique

L'`revoke-security-group-egress`exemple de commande suivant supprime la règle qui accorde l'accès au groupe de sécurité spécifié sur le port TCP 80.

```
aws ec2 revoke-security-group-egress \
    --group-id sg-026c12253ce15eff7 \
    --ip-permissions '[{"IpProtocol": "tcp", "FromPort": 443, "ToPort":
 443,"UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous RevokeSecurityGroupEgressà la section Référence des AWS CLI commandes.

`revoke-security-group-ingress`

L'exemple de code suivant montre comment utiliser`revoke-security-group-ingress`.

AWS CLI

Exemple 1 : pour supprimer une règle d'un groupe de sécurité

L'`revoke-security-group-ingress`exemple suivant supprime l'accès au port TCP 22 pour la plage d'`203.0.113.0/24`adresses du groupe de sécurité spécifié pour un VPC par défaut.

```
aws ec2 revoke-security-group-ingress \
    --group-name mySecurityGroup
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

Cette commande ne produit aucune sortie en cas de succès.

Pour plus d'informations, consultez la section Groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour supprimer une règle à l'aide des autorisations IP définies

L'`revoke-security-group-ingress`exemple suivant utilise le `ip-permissions` paramètre pour supprimer une règle entrante qui autorise le message ICMP `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (type 3, code 4).

```
aws ec2 revoke-security-group-ingress \
    --group-id sg-026c12253ce15eff7 \
    --ip-
permissions IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]
```

Cette commande ne produit aucune sortie en cas de succès.

Pour plus d'informations, consultez la section Groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous RevokeSecurityGroupIngressà la section Référence des AWS CLI commandes.

`run-instances`

L'exemple de code suivant montre comment utiliser`run-instances`.

AWS CLI

Exemple 1 : lancement d'une instance dans un sous-réseau par défaut

L'exemple `run-instances` suivant lance une instance unique de type `t2.micro` dans le sous-réseau par défaut de la région actuelle et l'associe au sous-réseau par défaut du VPC par défaut de la région. La paire de clés est facultative si vous ne prévoyez pas de vous connecter à votre instance via SSH (Linux) ou RDP (Windows).

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --key-name MyKeyPair
```

Sortie :

```
{
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0abcdef1234567890",
            "InstanceId": "i-1231231230abcdef0",
            "InstanceType": "t2.micro",
            "KeyName": "MyKeyPair",
            "LaunchTime": "2018-05-10T08:05:20.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-2a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
            "PrivateIpAddress": "10.0.0.157",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-04a636d18e83cfacb",
```

```
            "VpcId": "vpc-1234567890abcdef0",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "",
            "EbsOptimized": false,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2018-05-10T08:05:20.000Z",
                        "AttachmentId": "eni-attach-0e325c07e928a0405",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching"
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupName": "MySecurityGroup",
                            "GroupId": "sg-0598c7d356eba48d7"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "0a:ab:58:e0:67:e2",
                    "NetworkInterfaceId": "eni-0c0a29997760baee7",
                    "OwnerId": "123456789012",
                    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
                    "PrivateIpAddress": "10.0.0.157",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                            "PrivateIpAddress": "10.0.0.157"
                        }
                    ],
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-04a636d18e83cfacb",
                    "VpcId": "vpc-1234567890abcdef0",
                    "InterfaceType": "interface"
                }
            ],
            "RootDeviceName": "/dev/xvda",
```

```
            "RootDeviceType": "ebs",
            "SecurityGroups": [
                {
                    "GroupName": "MySecurityGroup",
                    "GroupId": "sg-0598c7d356eba48d7"
                }
            ],
            "SourceDestCheck": true,
            "StateReason": {
                "Code": "pending",
                "Message": "pending"
            },
            "Tags": [],
            "VirtualizationType": "hvm",
            "CpuOptions": {
                "CoreCount": 1,
                "ThreadsPerCore": 1
            },
            "CapacityReservationSpecification": {
                "CapacityReservationPreference": "open"
            },
            "MetadataOptions": {
                "State": "pending",
                "HttpTokens": "optional",
                "HttpPutResponseHopLimit": 1,
                "HttpEndpoint": "enabled"
            }
        }
    ],
    "OwnerId": "123456789012",
    "ReservationId": "r-02a3f596d91211712"
}
```

Exemple 2 : lancement d'une instance dans un sous-réseau autre que celui par défaut et ajout d'une adresse IP publique

L'exemple `run-instances` suivant demande une adresse IP publique pour une instance que vous lancez dans un sous-réseau autre que celui par défaut. L'instance est associée au groupe de sécurité spécifié.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
```

```
    --subnet-id subnet-08fc749671b2d077c \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --associate-public-ip-address \
    --key-name MyKeyPair
```

Pour obtenir un exemple de la manière de procéder pour `run-instances`, reportez-vous à l'exemple 1.

Exemple 3 : lancement d'une instance avec des volumes supplémentaires

L'exemple `run-instances` suivant utilise un mappage de périphérique de stockage en mode bloc, spécifié dans le fichier mapping.json, pour attacher des volumes supplémentaires au lancement. Un mappage de périphérique de stockage en mode bloc peut spécifier des volumes EBS, des volumes de stockage d'instance ou à la fois des volumes EBS et des volumes de stockage d'instance.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --subnet-id subnet-08fc749671b2d077c \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --key-name MyKeyPair \
    --block-device-mappings file://mapping.json
```

Contenu de `mapping.json`. Cet exemple ajoute `/dev/sdh`, un volume EBS vide d'une taille de 100 Gio.

```
[
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 100
        }
    }
]
```

Contenu de `mapping.json`. Cet exemple ajoute `ephemeral1` en tant que volume de stockage d'instances.

```
[
```

```
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral1"
    }
]
```

Pour obtenir un exemple de la manière de procéder pour `run-instances`, reportez-vous à l'exemple 1.

Pour plus d'informations sur les mappages d'appareils en mode bloc, consultez la section Mappage d'appareils en mode bloc dans le guide de EC2 l'utilisateur Amazon.

Exemple 4 : lancement d'une instance et ajout de balises lors de sa création

L'exemple `run-instances` suivant ajoute une balise avec une clé `webserver` et une valeur `production` à l'instance. La commande applique également une balise avec une clé de `cost-center` et une valeur de `cc123` à n'importe quel volume EBS qui est créé (dans ce cas, le volume racine).

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --tag-specifications
 'ResourceType=instance,Tags=[{Key=webserver,Value=production}]'
 'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

Pour obtenir un exemple de la manière de procéder pour `run-instances`, reportez-vous à l'exemple 1.

Exemple 5 : lancement d'une instance avec des données utilisateur

L'exemple `run-instances` suivant transmet les données de l'utilisateur dans un fichier appelé `my_script.txt` qui contient un script de configuration pour votre instance. Le script s'exécute au lancement.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
```

```
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --user-data file://my_script.txt
```

Pour obtenir un exemple de la manière de procéder pour `run-instances`, reportez-vous à l'exemple 1.

Pour plus d'informations sur les données utilisateur de l'instance, consultez la section Utilisation des données utilisateur de l'instance dans le guide de EC2 l'utilisateur Amazon.

Exemple 6 : lancement d'une instance à performances extensibles

L'exemple `run-instances` suivant lance une instance t2.micro avec l'option de crédit `unlimited`. Lorsque vous lancez une instance T2, si vous ne spécifiez pas `--credit-specification`, l'option de crédit `standard` est utilisée par défaut. Lorsque vous lancez une instance T3, l'option de crédit `unlimited` est utilisée par défaut.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --credit-specification CpuCredits=unlimited
```

Pour obtenir un exemple de la manière de procéder pour `run-instances`, reportez-vous à l'exemple 1.

Pour plus d'informations sur les instances de performance burstable, consultez la section Instances de performance burstable dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous RunInstancesà la section Référence des AWS CLI commandes.

`run-scheduled-instances`

L'exemple de code suivant montre comment utiliser`run-scheduled-instances`.

AWS CLI

Pour lancer une instance planifiée

Cet exemple lance l'instance planifiée spécifiée dans un VPC.

Commande :

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Spécification.json de lancement :

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "InstanceType": "c4.large",
  "NetworkInterfaces": [
    {
        "DeviceIndex": 0,
        "SubnetId": "subnet-12345678",
        "AssociatePublicIpAddress": true,
        "Groups": ["sg-12345678"]
    }
  ],
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

Sortie :

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

Cet exemple lance l'instance planifiée spécifiée dans EC2 -Classic.

Commande :

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Spécification.json de lancement :

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

Sortie :

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous RunScheduledInstancesà la section Référence des AWS CLI commandes.

`search-local-gateway-routes`

L'exemple de code suivant montre comment utiliser`search-local-gateway-routes`.

AWS CLI

Pour rechercher des itinéraires dans une table de routage de passerelle locale

L'`search-local-gateway-routes`exemple suivant recherche des itinéraires statiques dans la table de routage de passerelle locale spécifiée.

```
aws ec2 search-local-gateway-routes \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --filters "Name=type,Values=static"
```

Sortie :

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous SearchLocalGatewayRoutesà la section Référence des AWS CLI commandes.

`search-transit-gateway-multicast-groups`

L'exemple de code suivant montre comment utiliser`search-transit-gateway-multicast-groups`.

AWS CLI

Pour rechercher un ou plusieurs groupes de multidiffusion de passerelles de transit et renvoyer les informations d'appartenance au groupe

L'`search-transit-gateway-multicast-groups`exemple suivant renvoie l'appartenance au groupe de multidiffusion spécifié.

```
aws ec2 search-transit-gateway-multicast-groups \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

Sortie :

```
{
    "MulticastGroups": [
        {
            "GroupIpAddress": "224.0.1.0",
```

```
            "TransitGatewayAttachmentId": "tgw-attach-0372e72386EXAMPLE",
            "SubnetId": "subnet-0187aff814EXAMPLE",
            "ResourceId": "vpc-0065acced4EXAMPLE",
            "ResourceType": "vpc",
            "NetworkInterfaceId": "eni-03847706f6EXAMPLE",
            "GroupMember": false,
            "GroupSource": true,
            "SourceType": "static"
        }
    ]
}
```

Pour plus d'informations, consultez la section Multidiffusion sur les passerelles de transport en commun dans le Guide des passerelles de transport en commun.

- Pour plus de détails sur l'API, reportez-vous SearchTransitGatewayMulticastGroupsà la section Référence des AWS CLI commandes.

`search-transit-gateway-routes`

L'exemple de code suivant montre comment utiliser`search-transit-gateway-routes`.

AWS CLI

Pour rechercher des itinéraires dans la table de routage de la passerelle de transit spécifiée

L'`search-transit-gateway-routes`exemple suivant renvoie toutes les routes qui sont de type `static` dans la table de routage spécifiée.

```
aws ec2 search-transit-gateway-routes \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --filters "Name=type,Values=static"
```

Sortie :

```
{
    "Routes": [
        {
            "DestinationCidrBlock": "10.0.2.0/24",
            "TransitGatewayAttachments": [
                {
                    "ResourceId": "vpc-4EXAMPLE",
```

```
                    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                    "ResourceType": "vpc"
                }
            ],
            "Type": "static",
            "State": "active"
        },
        {
            "DestinationCidrBlock": "10.1.0.0/24",
            "TransitGatewayAttachments": [
                {
                    "ResourceId": "vpc-4EXAMPLE",
                    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                    "ResourceType": "vpc"
                }
            ],
            "Type": "static",
            "State": "active"
        }
    ],
    "AdditionalRoutesAvailable": false
}
```

Pour plus d'informations, consultez les tableaux d'itinéraires des passerelles de transit dans le guide des passerelles de transit.

- Pour plus de détails sur l'API, reportez-vous SearchTransitGatewayRoutesà la section Référence des AWS CLI commandes.

`send-diagnostic-interrupt`

L'exemple de code suivant montre comment utiliser`send-diagnostic-interrupt`.

AWS CLI

Pour envoyer une interruption de diagnostic

L'`send-diagnostic-interrupt`exemple suivant envoie une interruption de diagnostic à l'instance spécifiée.

```
aws ec2 send-diagnostic-interrupt \
    --instance-id i-1234567890abcdef0
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SendDiagnosticInterruptà la section Référence des AWS CLI commandes.

`start-instances`

L'exemple de code suivant montre comment utiliser`start-instances`.

AWS CLI

Pour démarrer une EC2 instance Amazon

L’exemple suivant affiche une instance basée sur Amazon EBS.

Commande :

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
    "StartingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

Pour plus d’informations, veuillez consulter la rubrique Arrêt et démarrage de votre instance dans le Guide de l’utilisateur Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous StartInstancesà la section Référence des AWS CLI commandes.

`start-network-insights-access-scope-analysis`

L'exemple de code suivant montre comment utiliser`start-network-insights-access-scope-analysis`.

AWS CLI

Pour démarrer une analyse de l'étendue d'accès à Network Insights

L'`start-network-insights-access-scope-analysis`exemple suivant lance l'analyse du périmètre dans votre AWS compte.

```
aws ec2 start-network-insights-access-scope-analysis \
    --region us-east-1 \
    --network-insights-access-scope-id nis-123456789111
```

Sortie :

```
{
    "NetworkInsightsAccessScopeAnalysis": {
        "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
        "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789222",
        "NetworkInsightsAccessScopeId": "nis-123456789111",
        "Status": "running",
        "StartDate": "2022-01-26T00:47:06.814000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Mise en route avec Network Access Analyzer à l'aide de la AWS CLI dans le Guide de l'analyseur d'accès réseau.

- Pour plus de détails sur l'API, reportez-vous StartNetworkInsightsAccessScopeAnalysisà la section Référence des AWS CLI commandes.

`start-network-insights-analysis`

L'exemple de code suivant montre comment utiliser`start-network-insights-analysis`.

AWS CLI

Pour analyser un chemin

L'`start-network-insights-analysis`exemple suivant analyse le chemin entre la source et la destination. Pour afficher les résultats de l'analyse du chemin, utilisez la `describe-network-insights-analyses` commande.

```
aws ec2 start-network-insights-analysis \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

Sortie :

```
{
    "NetworkInsightsAnalysis": {
        "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
        "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-analysis/nia-02207aa13eb480c7a",
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "StartDate": "2021-01-20T22:58:37.495Z",
        "Status": "running"
    }
}
```

Pour plus d'informations, consultez Getting started using the AWS CLI dans le guide Reachability Analyzer.

- Pour plus de détails sur l'API, reportez-vous StartNetworkInsightsAnalysisà la section Référence des AWS CLI commandes.

`start-vpc-endpoint-service-private-dns-verification`

L'exemple de code suivant montre comment utiliser`start-vpc-endpoint-service-private-dns-verification`.

AWS CLI

Pour lancer le processus de vérification DNS

L'`start-vpc-endpoint-service-private-dns-verification`exemple suivant lance le processus de vérification DNS pour le service de point de terminaison spécifié.

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \
    --service-id vpce-svc-071afff70666e61e0
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gérer les noms DNS dans le guide de AWS PrivateLink l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous StartVpcEndpointServicePrivateDnsVerificationà la section Référence des AWS CLI commandes.

`stop-instances`

L'exemple de code suivant montre comment utiliser`stop-instances`.

AWS CLI

Exemple 1 : pour arrêter une EC2 instance Amazon

L'exemple `stop-instances` suivant arrête l'instance Amazon EBS spécifiée.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
    "StoppingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Pour plus d'informations, veuillez consulter la rubrique Arrêt et démarrage de votre instance dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

Exemple 2 : pour mettre en veille prolongée une instance Amazon EC2

L'exemple `stop-instances` suivant met en hibernation l'instance soutenue par Amazon EBS si l'instance est activée pour l'hibernation et remplit les conditions préalables à l'hibernation. L'instance est arrêtée après avoir été mise en hibernation.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

Sortie :

```
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-1234567890abcdef0",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Pour plus d'informations, veuillez consulter la rubrique Mise en veille prolongée de votre instance à la demande Linux dans le Guide de l'utilisateur Amazon Elastic Compute Cloud.

- Pour plus de détails sur l'API, reportez-vous StopInstancesà la section Référence des AWS CLI commandes.

`terminate-client-vpn-connections`

L'exemple de code suivant montre comment utiliser`terminate-client-vpn-connections`.

AWS CLI

Pour mettre fin à une connexion à un point de terminaison VPN du Client

L'`terminate-client-vpn-connections`exemple suivant met fin à la connexion spécifiée au point de terminaison VPN du Client.

```
aws ec2 terminate-client-vpn-connections \
    --client-vpn-endpoint-id vpn-endpoint-123456789123abcde \
    --connection-id cvpn-connection-04edd76f5201e0cb8
```

Sortie :

```
{
    "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",
    "ConnectionStatuses": [
        {
            "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",
            "PreviousStatus": {
                "Code": "active"
            },
            "CurrentStatus": {
                "Code": "terminating"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Connexions client dans le Guide de l'administrateur du VPN AWS client.

- Pour plus de détails sur l'API, reportez-vous TerminateClientVpnConnectionsà la section Référence des AWS CLI commandes.

`terminate-instances`

L'exemple de code suivant montre comment utiliser`terminate-instances`.

AWS CLI

Pour mettre fin à une EC2 instance Amazon

Cet exemple résilie l'instance spécifiée.

Commande :

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des EC2 instances Amazon dans le guide de l'utilisateur de l'interface de ligne de AWS commande.

- Pour plus de détails sur l'API, reportez-vous TerminateInstancesà la section Référence des AWS CLI commandes.

`unassign-ipv6-addresses`

L'exemple de code suivant montre comment utiliser`unassign-ipv6-addresses`.

AWS CLI

Pour annuler l'attribution d'une IPv6 adresse depuis une interface réseau

Cet exemple annule l'attribution de l' IPv6 adresse spécifiée à l'interface réseau spécifiée.

Commande :

```
aws ec2 unassign-ipv6-addresses --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

Sortie :

```
{
  "NetworkInterfaceId": "eni-23c49b68",
  "UnassignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071"
  ]
}
```

- Pour plus de détails sur l'API, consultez la section UnassignIpv6Addresses dans AWS CLI Command Reference.

`unassign-private-ip-addresses`

L'exemple de code suivant montre comment utiliser`unassign-private-ip-addresses`.

AWS CLI

Pour annuler l'attribution d'une adresse IP privée secondaire depuis une interface réseau

Cet exemple annule l'attribution de l'adresse IP privée spécifiée à l'interface réseau spécifiée. Si la commande réussit, aucune sortie n'est renvoyée.

Commande :

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

- Pour plus de détails sur l'API, reportez-vous UnassignPrivateIpAddressesà la section Référence des AWS CLI commandes.

`unassign-private-nat-gateway-address`

L'exemple de code suivant montre comment utiliser`unassign-private-nat-gateway-address`.

AWS CLI

Pour annuler l'attribution d'une adresse IP privée à votre passerelle NAT privée

L'`unassign-private-nat-gateway-address`exemple suivant annule l'attribution de l'adresse IP spécifiée à la passerelle NAT privée spécifiée.

```
aws ec2 unassign-private-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --private-ip-addresses 10.0.20.197
```

Sortie :

```
{
    "NatGatewayId": "nat-0ee3edd182361f662",
    "NatGatewayAddresses": [
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "PrivateIp": "10.0.20.197",
            "IsPrimary": false,
            "Status": "unassigning"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter NAT Gateways (Passerelles NAT) dans le Guide de l'utilisateur Amazon VPC.

- Pour plus de détails sur l'API, reportez-vous UnassignPrivateNatGatewayAddressà la section Référence des AWS CLI commandes.

`unlock-snapshot`

L'exemple de code suivant montre comment utiliser`unlock-snapshot`.

AWS CLI

Pour déverrouiller un instantané

L'`unlock-snapshot`exemple suivant déverrouille le cliché spécifié.

```
aws ec2 unlock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d
```

Sortie :

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d"
```

```
}
```

Pour plus d'informations, consultez Snapshot Lock dans le guide de l'utilisateur d'Amazon EBS.

- Pour plus de détails sur l'API, reportez-vous UnlockSnapshotà la section Référence des AWS CLI commandes.

`unmonitor-instances`

L'exemple de code suivant montre comment utiliserunmonitor-instances.

AWS CLI

Désactivation de la surveillance détaillée d'une instance

Cet exemple de commande désactive la surveillance détaillée de l'instance spécifiée.

Commande :

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

Sortie :

```
{
  "InstanceMonitorings": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "Monitoring": {
              "State": "disabling"
          }
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous UnmonitorInstancesà la section Référence des AWS CLI commandes.

`update-security-group-rule-descriptions-egress`

L'exemple de code suivant montre comment utiliserupdate-security-group-rule-descriptions-egress.

AWS CLI

Pour mettre à jour la description d'une règle de groupe de sécurité sortant

L'`update-security-group-rule-descriptions-egress`exemple suivant met à jour la description de la règle du groupe de sécurité pour le port et la plage d' IPv4 adresses spécifiés. La description « `Outbound HTTP access to server 2` » remplace toute description existante de la règle.

```
aws ec2 update-security-group-rule-descriptions-egress \
    --group-id sg-02f0d35a850ba727f \
    --ip-permissions
 IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=203.0.113.0/24,Description="Outbound
 HTTP access to server 2"}]
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez la section Règles relatives aux groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateSecurityGroupRuleDescriptionsEgressà la section Référence des AWS CLI commandes.

`update-security-group-rule-descriptions-ingress`

L'exemple de code suivant montre comment utiliser`update-security-group-rule-descriptions-ingress`.

AWS CLI

Exemple 1 : pour mettre à jour la description d'une règle de groupe de sécurité entrant avec une source CIDR

L'`update-security-group-rule-descriptions-ingress`exemple suivant met à jour la description de la règle du groupe de sécurité pour le port et la plage d' IPv4 adresses spécifiés. La description « `SSH access from ABC office` » remplace toute description existante de la règle.

```
aws ec2 update-security-group-rule-descriptions-ingress \
    --group-id sg-02f0d35a850ba727f \
    --ip-permissions
 IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='[{CidrIp=203.0.113.0/16,Description="SSH
 access from corpnet"}]'
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez la section Règles relatives aux groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

Exemple 2 : pour mettre à jour la description d'une règle de groupe de sécurité entrant avec une source de liste de préfixes

L'`update-security-group-rule-descriptions-ingress`exemple suivant met à jour la description de la règle du groupe de sécurité pour le port et la liste de préfixes spécifiés. La description « `SSH access from ABC office` » remplace toute description existante de la règle.

```
aws ec2 update-security-group-rule-descriptions-ingress \
    --group-id sg-02f0d35a850ba727f \
    --ip-permissions
 IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds='[{PrefixListId=pl-12345678,Description=
 access from corpnet"}]'
```

Sortie :

```
{
    "Return": true
}
```

Pour plus d'informations, consultez la section Règles relatives aux groupes de sécurité dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateSecurityGroupRuleDescriptionsIngressà la section Référence des AWS CLI commandes.

`withdraw-byoip-cidr`

L'exemple de code suivant montre comment utiliser`withdraw-byoip-cidr`.

AWS CLI

Pour arrêter de faire de la publicité pour une plage d'adresses

L'`withdraw-byoip-cidr`exemple suivant arrête de publier la plage d'adresses spécifiée.

```
aws ec2 withdraw-byoip-cidr
    --cidr 203.0.113.25/24
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
        "State": "advertised"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous WithdrawByoipCidrà la section Référence des AWS CLI commandes.

Exemples d'Amazon EC2 Instance Connect utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon EC2 Instance Connect.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`send-ssh-public-key`

L'exemple de code suivant montre comment utiliser`send-ssh-public-key`.

AWS CLI

Pour envoyer une clé publique SSH à une instance

L'`send-ssh-public-key`exemple suivant envoie la clé publique SSH spécifiée à l'instance spécifiée. La clé est utilisée pour authentifier l'utilisateur spécifié.

```
aws ec2-instance-connect send-ssh-public-key \
    --instance-id i-1234567890abcdef0 \
    --instance-os-user ec2-user \
    --availability-zone us-east-2b \
    --ssh-public-key file://path/my-rsa-key.pub
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SendSshPublicKeyà la section Référence des AWS CLI commandes.

Exemples d'Amazon ECR utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon ECR.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-check-layer-availability`

L'exemple de code suivant montre comment utiliser`batch-check-layer-availability`.

AWS CLI

Pour vérifier la disponibilité d'une couche

L'`batch-check-layer-availability`exemple suivant vérifie la disponibilité d'une couche avec le condensé `sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed` dans le `cluster-autoscaler` référentiel.

```
aws ecr batch-check-layer-availability \
    --repository-name cluster-autoscaler \
    --layer-
digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

Sortie :

```
{
    "layers": [
        {
            "layerDigest":
 "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",
            "layerAvailability": "AVAILABLE",
            "layerSize": 2777,
            "mediaType": "application/vnd.docker.container.image.v1+json"
        }
    ],
    "failures": []
}
```

- Pour plus de détails sur l'API, reportez-vous BatchCheckLayerAvailabilityà la section Référence des AWS CLI commandes.

`batch-delete-image`

L'exemple de code suivant montre comment utiliser`batch-delete-image`.

AWS CLI

Exemple 1 : pour supprimer une image

L'`batch-delete-image`exemple suivant supprime une image avec le tag `precise` dans le référentiel spécifié dans le registre par défaut d'un compte.

```
aws ecr batch-delete-image \
    --repository-name ubuntu \
    --image-ids imageTag=precise
```

Sortie :

```
{
    "failures": [],
    "imageIds": [
        {
            "imageTag": "precise",
            "imageDigest":
 "sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"
        }
    ]
}
```

Exemple 2 : pour supprimer plusieurs images

L'`batch-delete-image`exemple suivant supprime toutes les images étiquetées avec `prod` et `team1` dans le référentiel spécifié.

```
aws ecr batch-delete-image \
    --repository-name MyRepository \
    --image-ids imageTag=prod imageTag=team1
```

Sortie :

```
{
    "imageIds": [
        {
            "imageDigest": "sha256:123456789012",
            "imageTag": "prod"
        },
        {
```

```
            "imageDigest": "sha256:567890121234",
            "imageTag": "team1"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez Supprimer une image dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous BatchDeleteImageà la section Référence des AWS CLI commandes.

`batch-get-image`

L'exemple de code suivant montre comment utiliser`batch-get-image`.

AWS CLI

Exemple 1 : Pour obtenir une image

L'`batch-get-image`exemple suivant permet d'obtenir une image avec la balise `v1.13.6` dans un référentiel appelé `cluster-autoscaler` dans le registre par défaut d'un compte.

```
aws ecr batch-get-image \
    --repository-name cluster-autoscaler \
    --image-ids imageTag=v1.13.6
```

Sortie :

```
{
    "images": [
        {
            "registryId": "012345678910",
            "repositoryName": "cluster-autoscaler",
            "imageId": {
                "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
                "imageTag": "v1.13.6"
            },
            "imageManifest": "{\n   \"schemaVersion\": 2,\n
 \"mediaType\": \"application/vnd.docker.distribution.manifest.v2+json
```

```
\",\n   \"config\": {\n      \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",\n      \"size\": 2777,\n      \"digest
\": \"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
\"\n   },\n   \"layers\": [\n      {\n         \"mediaType
\": \"application/vnd.docker.image.rootfs.diff.tar.gzip
\",\n         \"size\": 17743696,\n         \"digest\":
 \"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffeb5a85f900caaf6e1\"\n
      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 2565026,\n
 \"digest\":
 \"sha256:8c8a779d3a537b767ae1091fe6e00c2590afd16767aa6096d1b318d75494819f
\"\n      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 28005981,\n
 \"digest\":
 \"sha256:c44ba47496991c9982ee493b47fd25c252caabf2b4ae7dd679c9a27b6a3c8fb7\"\n
      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 775,\n         \"digest
\": \"sha256:e2c388b44226544363ca007be7b896bcce1baebea04da23cbd165eac30be650f\"\n
   }\n   ]\n}"
        }
    ],
    "failures": []
}
```

Exemple 2 : pour obtenir plusieurs images

L'`batch-get-image`exemple suivant affiche les détails de toutes les images étiquetées avec `prod` et `team1` dans le référentiel spécifié.

```
aws ecr batch-get-image \
    --repository-name MyRepository \
    --image-ids imageTag=prod imageTag=team1
```

Sortie :

```
{
    "images": [
        {
            "registryId": "123456789012",
            "repositoryName": "MyRepository",
            "imageId": {
                "imageDigest": "sha256:123456789012",
                "imageTag": "prod"
```

```
            },
            "imageManifest": "manifestExample1"
        },
        {
            "registryId": "567890121234",
            "repositoryName": "MyRepository",
            "imageId": {
                "imageDigest": "sha256:123456789012",
                "imageTag": "team1"
            },
            "imageManifest": "manifestExample2"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Images du guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous BatchGetImageà la section Référence des AWS CLI commandes.

`complete-layer-upload`

L'exemple de code suivant montre comment utiliser`complete-layer-upload`.

AWS CLI

Pour terminer le téléchargement d'une couche d'images

L'`complete-layer-upload`exemple suivant termine le téléchargement d'une couche d'image vers le `layer-test` référentiel.

```
aws ecr complete-layer-upload \
    --repository-name layer-test \
    --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
    --layer-digests 6cb64b8a-9378-0e33-2ab1-
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

Sortie :

```
{
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
```

```
    "layerDigest":
 "sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",
    "repositoryName": "layer-test",
    "registryId": "130757420319"
}
```

- Pour plus de détails sur l'API, reportez-vous CompleteLayerUploadà la section Référence des AWS CLI commandes.

`create-repository`

L'exemple de code suivant montre comment utiliser`create-repository`.

AWS CLI

Exemple 1 : pour créer un référentiel

L'`create-repository`exemple suivant crée un dépôt dans l'espace de noms spécifié dans le registre par défaut d'un compte.

```
aws ecr create-repository \
    --repository-name project-a/sample-repo
```

Sortie :

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo"
    }
}
```

Pour plus d'informations, consultez la section Création d'un référentiel dans le guide de l'utilisateur Amazon ECR.

Exemple 2 : pour créer un référentiel configuré avec l'immuabilité des balises d'image

L'`create-repository`exemple suivant crée un référentiel configuré pour l'immuabilité des balises dans le registre par défaut d'un compte.

```
aws ecr create-repository \
    --repository-name project-a/sample-repo \
    --image-tag-mutability IMMUTABLE
```

Sortie :

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo",
        "imageTagMutability": "IMMUTABLE"
    }
}
```

Pour plus d'informations, consultez la section Mutabilité des balises d'image dans le guide de l'utilisateur Amazon ECR.

Exemple 3 : pour créer un référentiel configuré avec une configuration de numérisation

L'`create-repository`exemple suivant crée un référentiel configuré pour effectuer une analyse de vulnérabilité lors du transfert d'image dans le registre par défaut d'un compte.

```
aws ecr create-repository \
    --repository-name project-a/sample-repo \
    --image-scanning-configuration scanOnPush=true
```

Sortie :

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo",
        "imageScanningConfiguration": {
            "scanOnPush": true
        }
    }
}
```

Pour plus d'informations, consultez la section Numérisation d'images dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous CreateRepositoryà la section Référence des AWS CLI commandes.

`delete-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`delete-lifecycle-policy`.

AWS CLI

Pour supprimer la politique de cycle de vie d'un référentiel

L'`delete-lifecycle-policy`exemple suivant supprime la politique de cycle de vie du `hello-world` référentiel.

```
aws ecr delete-lifecycle-policy \
    --repository-name hello-world
```

Sortie :

```
{
    "registryId": "012345678910",
    "repositoryName": "hello-world",
    "lifecyclePolicyText": "{\"rules\":[{\"rulePriority\":1,\"description\":\"Remove
 untagged images.\",\"selection\":{\"tagStatus\":\"untagged\",\"countType\":
\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":10},\"action\":{\"type
\":\"expire\"}}]}",
    "lastEvaluatedAt": 0.0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteLifecyclePolicyà la section Référence des AWS CLI commandes.

`delete-repository-policy`

L'exemple de code suivant montre comment utiliser`delete-repository-policy`.

AWS CLI

Pour supprimer la politique de référentiel d'un référentiel

L'`delete-repository-policy`exemple suivant supprime la politique de référentiel pour le `cluster-autoscaler` référentiel.

```
aws ecr delete-repository-policy \
    --repository-name cluster-autoscaler
```

Sortie :

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteRepositoryPolicyà la section Référence des AWS CLI commandes.

`delete-repository`

L'exemple de code suivant montre comment utiliser`delete-repository`.

AWS CLI

Pour supprimer un référentiel

L'`delete-repository`exemple de force de commande suivant supprime le référentiel spécifié dans le registre par défaut d'un compte. L'`--force`indicateur est obligatoire si le référentiel contient des images.

```
aws ecr delete-repository \
    --repository-name ubuntu \
    --force
```

Sortie :

```
{
    "repository": {
```

```
        "registryId": "123456789012",
        "repositoryName": "ubuntu",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"
    }
}
```

Pour plus d'informations, consultez Supprimer un référentiel dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous DeleteRepositoryà la section Référence des AWS CLI commandes.

`describe-image-scan-findings`

L'exemple de code suivant montre comment utiliser`describe-image-scan-findings`.

AWS CLI

Pour décrire les résultats de numérisation d'une image

L'`describe-image-scan-findings`exemple suivant renvoie les résultats de numérisation d'une image à l'aide du condensé d'image du référentiel spécifié dans le registre par défaut d'un compte.

```
aws ecr describe-image-scan-findings \
    --repository-name sample-repo \
    --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

Sortie :

```
{
    "imageScanFindings": {
      "findings": [
          {
              "name": "CVE-2019-5188",
              "description": "A code execution vulnerability exists in the directory
 rehashing functionality of E2fsprogs e2fsck 1.45.4. A specially crafted ext4
 directory can cause an out-of-bounds write on the stack, resulting in code
 execution. An attacker can corrupt a partition to trigger this vulnerability.",
              "uri": "http://people.ubuntu.com/~ubuntu-security/cve/CVE-2019-5188",
```

```
                "severity": "MEDIUM",
                "attributes": [
                    {
                        "key": "package_version",
                        "value": "1.44.1-1ubuntu1.1"
                    },
                    {
                        "key": "package_name",
                        "value": "e2fsprogs"
                    },
                    {
                        "key": "CVSS2_VECTOR",
                        "value": "AV:L/AC:L/Au:N/C:P/I:P/A:P"
                    },
                    {
                        "key": "CVSS2_SCORE",
                        "value": "4.6"
                    }
                ]
            }
        ],
        "imageScanCompletedAt": 1579839105.0,
        "vulnerabilitySourceUpdatedAt": 1579811117.0,
        "findingSeverityCounts": {
            "MEDIUM": 1
        }
    },
    "registryId": "123456789012",
    "repositoryName": "sample-repo",
    "imageId": {
        "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
    },
    "imageScanStatus": {
        "status": "COMPLETE",
        "description": "The scan was completed successfully."
    }
}
```

Pour plus d'informations, consultez la section Numérisation d'images dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous DescribeImageScanFindingsà la section Référence des AWS CLI commandes.

`describe-images`

L'exemple de code suivant montre comment utiliser`describe-images`.

AWS CLI

Pour décrire une image dans un référentiel

L'`describe-images`exemple suivant affiche les détails d'une image dans le `cluster-autoscaler` référentiel avec la balise. `v1.13.6`

```
aws ecr describe-images \
    --repository-name cluster-autoscaler \
    --image-ids imageTag=v1.13.6
```

Sortie :

```
{
    "imageDetails": [
        {
            "registryId": "012345678910",
            "repositoryName": "cluster-autoscaler",
            "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
            "imageTags": [
                "v1.13.6"
            ],
            "imageSizeInBytes": 48318255,
            "imagePushedAt": 1565128275.0
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeImagesà la section Référence des AWS CLI commandes.

`describe-repositories`

L'exemple de code suivant montre comment utiliser`describe-repositories`.

AWS CLI

Pour décrire les référentiels d'un registre

Cet exemple décrit les référentiels du registre par défaut d'un compte.

Commande :

```
aws ecr describe-repositories
```

Sortie :

```
{
    "repositories": [
        {
            "registryId": "012345678910",
            "repositoryName": "ubuntu",
            "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/ubuntu"
        },
        {
            "registryId": "012345678910",
            "repositoryName": "test",
            "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/test"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeRepositoriesà la section Référence des AWS CLI commandes.

`get-authorization-token`

L'exemple de code suivant montre comment utiliser`get-authorization-token`.

AWS CLI

Pour obtenir un jeton d'autorisation pour votre registre par défaut

L'`get-authorization-token`exemple de commande suivant permet d'obtenir un jeton d'autorisation pour votre registre par défaut.

```
aws ecr get-authorization-token
```

Sortie :

```
{
    "authorizationData": [
        {
            "authorizationToken": "QVdTOkN...",
            "expiresAt": 1448875853.241,
            "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetAuthorizationTokenà la section Référence des AWS CLI commandes.

`get-download-url-for-layer`

L'exemple de code suivant montre comment utiliser`get-download-url-for-layer`.

AWS CLI

Pour obtenir l'URL de téléchargement d'une couche

L'`get-download-url-for-layer`exemple suivant affiche l'URL de téléchargement d'une couche contenant le condensé `sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed` dans le `cluster-autoscaler` référentiel.

```
aws ecr get-download-url-for-layer \
    --repository-name cluster-autoscaler  \
    --layer-
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

Sortie :

```
{
    "downloadUrl": "https://prod-us-west-2-starport-layer-bucket.s3.us-
west-2.amazonaws.com/e501-012345678910-9cb60dc0-7284-5643-3987-
da6dac0465f0/04620aac-66a5-4167-8232-55ee7ef6d565?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20190814T220617Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-
Amz-Credential=AKIA32P3D2JDNMVAJLGF%2F20190814%2Fus-west-2%2Fs3%2Faws4_request&X-
Amz-Signature=9161345894947a1672467a0da7a1550f2f7157318312fe4941b59976239c3337",
```

```
    "layerDigest":
 "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed"
}
```

- Pour plus de détails sur l'API, reportez-vous GetDownloadUrlForLayerà la section Référence des AWS CLI commandes.

`get-lifecycle-policy-preview`

L'exemple de code suivant montre comment utiliser`get-lifecycle-policy-preview`.

AWS CLI

Pour récupérer les détails d'un aperçu de la politique de cycle de vie

L'`get-lifecycle-policy-preview`exemple suivant extrait le résultat d'un aperçu de la politique de cycle de vie pour le référentiel spécifié dans le registre par défaut d'un compte.

Commande :

```
aws ecr get-lifecycle-policy-preview \
    --repository-name "project-a/amazon-ecs-sample"
```

Sortie :

```
{
    "registryId": "012345678910",
    "repositoryName": "project-a/amazon-ecs-sample",
    "lifecyclePolicyText": "{\n    \"rules\": [\n        {\n
 \"rulePriority\": 1,\n            \"description\": \"Expire images older than 14
 days\",\n            \"selection\": {\n                \"tagStatus\": \"untagged\",
\n                \"countType\": \"sinceImagePushed\",\n                \"countUnit
\": \"days\",\n                \"countNumber\": 14\n            },\n
 \"action\": {\n                \"type\": \"expire\"\n            }\n        }\n
  ]\n}\n",
    "status": "COMPLETE",
    "previewResults": [],
    "summary": {
        "expiringImageTotalCount": 0
    }
}
```

Pour plus d'informations, consultez les politiques de cycle de vie dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous GetLifecyclePolicyPreviewà la section Référence des AWS CLI commandes.

`get-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`get-lifecycle-policy`.

AWS CLI

Pour récupérer une politique de cycle de vie

L'`get-lifecycle-policy`exemple suivant affiche les détails de la politique de cycle de vie du référentiel spécifié dans le registre par défaut du compte.

```
aws ecr get-lifecycle-policy \
    --repository-name "project-a/amazon-ecs-sample"
```

Sortie :

```
{
     "registryId": "123456789012",
     "repositoryName": "project-a/amazon-ecs-sample",
     "lifecyclePolicyText": "{\"rules\":[{\"rulePriority\":1,\"description\":
\"Expire images older than 14 days\",\"selection\":{\"tagStatus\":\"untagged\",
\"countType\":\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":14},
\"action\":{\"type\":\"expire\"}}]}",
     "lastEvaluatedAt": 1504295007.0
}
```

Pour plus d'informations, consultez les politiques de cycle de vie dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous GetLifecyclePolicyà la section Référence des AWS CLI commandes.

`get-login-password`

L'exemple de code suivant montre comment utiliser`get-login-password`.

AWS CLI

Pour récupérer un mot de passe afin de s'authentifier auprès d'un registre

Ce qui suit `get-login-password` affiche un mot de passe que vous pouvez utiliser avec le client conteneur de votre choix pour vous authentifier auprès de n'importe quel registre Amazon ECR auquel votre principal IAM a accès.

```
aws ecr get-login-password
```

Sortie :

```
<password>
```

Pour une utilisation avec la CLI Docker, redirigez la sortie de la `get-login-password` commande vers la `docker login` commande. Lorsque vous récupérez le mot de passe, assurez-vous de spécifier la même région que celle dans laquelle se trouve votre registre Amazon ECR.

```
aws ecr get-login-password \
    --region <region>
 \
| docker login \
    --username AWS \
    --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Pour plus d'informations, consultez la section Authentification du registre dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous GetLoginPasswordà la section Référence des AWS CLI commandes.

`get-login`

L'exemple de code suivant montre comment utiliser`get-login`.

AWS CLI

Pour récupérer une commande de connexion Docker dans votre registre par défaut

Cet exemple imprime une commande que vous pouvez utiliser pour vous connecter à votre registre Amazon ECR par défaut.

Commande :

```
aws ecr get-login
```

Sortie :

```
docker login -u AWS -p <password> -e none https://
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Pour vous connecter au registre d'un autre compte

Cet exemple imprime une ou plusieurs commandes que vous pouvez utiliser pour vous connecter aux registres Amazon ECR associés à d'autres comptes.

Commande :

```
aws ecr get-login --registry-ids 012345678910 023456789012
```

Sortie :

```
docker login -u <username> -p <token-1> -e none <endpoint-1>
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- Pour plus de détails sur l'API, reportez-vous GetLoginà la section Référence des AWS CLI commandes.

`get-repository-policy`

L'exemple de code suivant montre comment utiliser`get-repository-policy`.

AWS CLI

Pour récupérer la politique de dépôt d'un référentiel

L'`get-repository-policy`exemple suivant affiche les détails de la politique de référentiel pour le `cluster-autoscaler` référentiel.

```
aws ecr get-repository-policy \
    --repository-name cluster-autoscaler
```

Sortie :

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- Pour plus de détails sur l'API, reportez-vous GetRepositoryPolicyà la section Référence des AWS CLI commandes.

`initiate-layer-upload`

L'exemple de code suivant montre comment utiliser`initiate-layer-upload`.

AWS CLI

Pour lancer le téléchargement d'une couche d'image

L'`initiate-layer-upload`exemple suivant lance le téléchargement d'une couche d'image vers le `layer-test` référentiel.

```
aws ecr initiate-layer-upload \
    --repository-name layer-test
```

Sortie :

```
{
    "partSize": 10485760,
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}
```

- Pour plus de détails sur l'API, reportez-vous InitiateLayerUploadà la section Référence des AWS CLI commandes.

`list-images`

L'exemple de code suivant montre comment utiliser`list-images`.

AWS CLI

Pour répertorier les images d'un référentiel

L'`list-images`exemple suivant affiche la liste des images du `cluster-autoscaler` référentiel.

```
aws ecr list-images \
    --repository-name cluster-autoscaler
```

Sortie :

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
            "imageTag": "v1.13.8"
        },
        {
            "imageDigest":
 "sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
            "imageTag": "v1.13.7"
        },
        {
            "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
            "imageTag": "v1.13.6"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListImagesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises du référentiel

L'`list-tags-for-resource`exemple suivant affiche la liste des balises associées au `hello-world` référentiel.

```
aws ecr list-tags-for-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world
```

Sortie :

```
{
    "tags": [
        {
            "Key": "Stage",
            "Value": "Integ"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-image-scanning-configuration`

L'exemple de code suivant montre comment utiliser`put-image-scanning-configuration`.

AWS CLI

Pour mettre à jour la configuration de numérisation d'images pour un référentiel

L'`put-image-scanning-configuration`exemple suivant met à jour la configuration de numérisation d'images pour le référentiel spécifié.

```
aws ecr put-image-scanning-configuration \
    --repository-name sample-repo \
    --image-scanning-configuration scanOnPush=true
```

Sortie :

```
{
```

```
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageScanningConfiguration": {
      "scanOnPush": true
    }
}
```

Pour plus d'informations, consultez la section Numérisation d'images dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous PutImageScanningConfigurationà la section Référence des AWS CLI commandes.

`put-image-tag-mutability`

L'exemple de code suivant montre comment utiliser`put-image-tag-mutability`.

AWS CLI

Pour mettre à jour le paramètre de mutabilité des balises d'image pour un référentiel

L'`put-image-tag-mutability`exemple suivant configure le référentiel spécifié pour l'immuabilité des balises. Cela empêche le remplacement de toutes les balises d'image du référentiel.

```
aws ecr put-image-tag-mutability \
    --repository-name hello-repository \
    --image-tag-mutability IMMUTABLE
```

Sortie :

```
{
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageTagMutability": "IMMUTABLE"
}
```

Pour plus d'informations, consultez la section Mutabilité des balises d'image dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous PutImageTagMutabilityà la section Référence des AWS CLI commandes.

`put-image`

L'exemple de code suivant montre comment utiliser`put-image`.

AWS CLI

Pour réétiqueter une image avec son manifeste

L'`put-image`exemple suivant crée une nouvelle balise dans le `hello-world` référentiel avec un manifeste d'image existant.

```
aws ecr put-image \
    --repository-name hello-world \
    --image-tag 2019.08 \
    --image-manifest file://hello-world.manifest.json
```

Contenu de `hello-world.manifest.json` :

```
{
    "schemaVersion": 2,
    "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
    "config": {
        "mediaType": "application/vnd.docker.container.image.v1+json",
        "size": 5695,
        "digest":
 "sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980"
    },
    "layers": [
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 39096921,
            "digest":
 "sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 57938,
            "digest":
 "sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 423,
```

```
            "digest":
"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 680,
            "digest":
"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 162,
            "digest":
"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 28268840,
            "digest":
"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 35369152,
            "digest":
"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 155,
            "digest":
"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 28737,
            "digest":
"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 190,
            "digest":
"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7"
```

```
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 28748,
            "digest":
 "sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee"
        }
    ]
}
```

Sortie :

```
{
    "image": {
        "registryId": "130757420319",
        "repositoryName": "hello-world",
        "imageId": {
            "imageDigest":
 "sha256:8ece96b74f87652876199d83bd107d0435a196133af383ac54cb82b6cc5283ae",
            "imageTag": "2019.08"
        },
        "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType
\": \"application/vnd.docker.distribution.manifest.v2+json
\",\n  \"config\": {\n    \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",\n    \"size\": 5695,\n    \"digest\":
 \"sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980\"\n
  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 39096921,\n      \"digest
\": \"sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295\"\n
  },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 57938,\n      \"digest
\": \"sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 423,\n      \"digest\":
 \"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 680,\n      \"digest\":
 \"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 162,\n      \"digest
\": \"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 28268840,\n      \"digest
```

```
\": \"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 35369152,\n      \"digest
\": \"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276\"\n
  },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 155,\n      \"digest\":
 \"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 28737,\n      \"digest\":
 \"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 190,\n      \"digest\":
 \"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 28748,\n      \"digest\":
 \"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee\"\n
  }\n  ]\n}\n"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutImageà la section Référence des AWS CLI commandes.

`put-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`put-lifecycle-policy`.

AWS CLI

Pour créer une politique de cycle de vie

L'`put-lifecycle-policy`exemple suivant crée une politique de cycle de vie pour le référentiel spécifié dans le registre par défaut d'un compte.

```
aws ecr put-lifecycle-policy \
    --repository-name "project-a/amazon-ecs-sample" \
    --lifecycle-policy-text "file://policy.json"
```

Contenu de `policy.json` :

```
{
   "rules": [
```

```
        {
            "rulePriority": 1,
            "description": "Expire images older than 14 days",
            "selection": {
                "tagStatus": "untagged",
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
```

Sortie :

```
{
   "registryId": "<aws_account_id>",
   "repositoryName": "project-a/amazon-ecs-sample",
   "lifecyclePolicyText": "{\"rules\":[{\"rulePriority\":1,\"description\":\"Expire
 images older than 14 days\",\"selection\":{\"tagStatus\":\"untagged\",\"countType
\":\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":14},\"action\":
{\"type\":\"expire\"}}]}"
}
```

Pour plus d'informations, consultez les politiques de cycle de vie dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous PutLifeCyclePolicyà la section Référence des AWS CLI commandes.

`set-repository-policy`

L'exemple de code suivant montre comment utiliser`set-repository-policy`.

AWS CLI

Pour définir la politique de dépôt d'un référentiel

L'`set-repository-policy`exemple suivant joint au référentiel une politique de `cluster-autoscaler` référentiel contenue dans un fichier.

```
aws ecr set-repository-policy \
    --repository-name cluster-autoscaler \
    --policy-text file://my-policy.json
```

Contenu de `my-policy.json` :

```
{
    "Version" : "2008-10-17",
    "Statement" : [
        {
            "Sid" : "allow public pull",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

Sortie :

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- Pour plus de détails sur l'API, reportez-vous SetRepositoryPolicyà la section Référence des AWS CLI commandes.

`start-image-scan`

L'exemple de code suivant montre comment utiliser`start-image-scan`.

AWS CLI

Pour démarrer une analyse de vulnérabilité d'une image

L'`start-image-scan`exemple suivant lance une numérisation d'image pour et spécifiée par le condensé d'image dans le référentiel spécifié.

```
aws ecr start-image-scan \
    --repository-name sample-repo \
    --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

Sortie :

```
{
   "registryId": "012345678910",
   "repositoryName": "sample-repo",
   "imageId": {
       "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
   },
   "imageScanStatus": {
       "status": "IN_PROGRESS"
   }
}
```

Pour plus d'informations, consultez la section Numérisation d'images dans le guide de l'utilisateur Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous StartImageScanà la section Référence des AWS CLI commandes.

`start-lifecycle-policy-preview`

L'exemple de code suivant montre comment utiliser`start-lifecycle-policy-preview`.

AWS CLI

Pour créer un aperçu de la politique de cycle de vie

L'`start-lifecycle-policy-preview`exemple suivant crée un aperçu de la politique de cycle de vie défini par un fichier JSON pour le référentiel spécifié.

```
aws ecr start-lifecycle-policy-preview \
    --repository-name "project-a/amazon-ecs-sample" \
    --lifecycle-policy-text "file://policy.json"
```

Contenu de `policy.json` :

```
{
   "rules": [
       {
           "rulePriority": 1,
           "description": "Expire images older than 14 days",
           "selection": {
               "tagStatus": "untagged",
               "countType": "sinceImagePushed",
               "countUnit": "days",
               "countNumber": 14
           },
           "action": {
               "type": "expire"
           }
       }
   ]
}
```

Sortie :

```
{
   "registryId": "012345678910",
   "repositoryName": "project-a/amazon-ecs-sample",
   "lifecyclePolicyText": "{\n    \"rules\": [\n        {\n
 \"rulePriority\": 1,\n            \"description\": \"Expire images older than 14
 days\",\n            \"selection\": {\n                \"tagStatus\": \"untagged\",
\n                \"countType\": \"sinceImagePushed\",\n                \"countUnit
\": \"days\",\n                \"countNumber\": 14\n            },\n
 \"action\": {\n                \"type\": \"expire\"\n            }\n        }\n
  ]\n}\n",
   "status": "IN_PROGRESS"
}
```

- Pour plus de détails sur l'API, reportez-vous StartLifecyclePolicyPreviewà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un dépôt

L'`tag-resource`exemple suivant définit une balise avec une clé `Stage` et une valeur `Integ` sur le `hello-world` référentiel.

```
aws ecr tag-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
    --tags Key=Stage,Value=Integ
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer le balisage d'un dépôt

L'`untag-resource`exemple suivant supprime la balise contenant la clé `Stage` du `hello-world` référentiel.

```
aws ecr untag-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
    --tag-keys Stage
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`upload-layer-part`

L'exemple de code suivant montre comment utiliser`upload-layer-part`.

AWS CLI

Pour télécharger une partie de couche

Ce qui suit `upload-layer-part` télécharge une partie de la couche d'image dans le `layer-test` référentiel.

```
aws ecr upload-layer-part \
    --repository-name layer-test \
    --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
    --part-first-byte 0 \
    --part-last-byte 8323314 \
    --layer-part-blob file:///var/lib/docker/image/overlay2/layerdb/sha256/
ff986b10a018b48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e/layer.b64
```

Sortie :

```
{
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
    "registryId": "012345678910",
    "lastByteReceived": 8323314,
    "repositoryName": "layer-test"
}
```

- Pour plus de détails sur l'API, reportez-vous UploadLayerPartà la section Référence des AWS CLI commandes.

Exemples d'Amazon ECR Public utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon ECR Public.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-delete-image`

L'exemple de code suivant montre comment utiliser`batch-delete-image`.

AWS CLI

Exemple 1 : pour supprimer une image à l'aide des identifiants de condensé d'image, l'image et toutes ses balises sont supprimées dans un référentiel d'un registre public

L'`batch-delete-image`exemple suivant supprime une image en spécifiant le condensé de l'image. :

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-
ids imageDigest=sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2
```

Sortie :

```
{
"imageIds": [
    {
        "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
        "imageTag": "latest"
    }
],
"failures": []
}
```

Pour plus d'informations, consultez la section Suppression d'une image dans un référentiel public dans le manuel Amazon ECR Public User Guide.

Exemple 2 : Supprimer une image en spécifiant la balise associée à l'image que vous souhaitez supprimer du référentiel.

L'`batch-delete-image`exemple suivant supprime une image en spécifiant la balise associée au référentiel d'images nommé `project-a/nginx-web-app` dans un registre public. Si vous n'avez qu'un seul tag et que vous exécutez cette commande, l'image sera supprimée. Sinon, si vous avez plusieurs balises pour la même image, spécifiez-en une, et seule la balise sera supprimée du référentiel, pas l'image.

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-ids imageTag=_temp
```

Sortie :

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
            "imageTag": "_temp"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Suppression d'une image dans un référentiel public dans le manuel Amazon ECR Public User Guide.

Exemple 3 : pour supprimer plusieurs images, vous pouvez spécifier plusieurs balises d'image ou résumés d'images dans la demande de dépôt dans un registre public.

L'`batch-delete-image`exemple suivant supprime plusieurs images d'un référentiel nommé project-a/ nginx-web-app en spécifiant plusieurs balises d'image ou résumés d'images dans la demande.

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-ids imageTag=temp2.0
  imageDigest=sha256:47ba980bc055353d9c0af89b1894f68faa43ca93856917b8406316be86f01278
```

Sortie :

```
{
     "imageIds": [
         {
             "imageDigest":
 "sha256:47ba980bc055353d9c0af89b1894f68faa43ca93856917b8406316be86f01278"
         },
         {
             "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
             "imageTag": "temp2.0"
         }
     ],
     "failures": []
 }
```

Pour plus d'informations, consultez la section Suppression d'une image dans un référentiel public dans le manuel Amazon ECR Public User Guide.

Exemple 4 : Pour supprimer une image dans Cross AWS Account à l'aide des identifiants registry-id et imagedigest, l'image et toutes ses balises sont supprimées dans un référentiel d'un registre public

L'`batch-delete-image`exemple suivant supprime une image en spécifiant le résumé de l'image dans le AWS compte croisé. :

```
aws ecr-public batch-delete-image \
    --registry-id 123456789098 \
    --repository-name project-a/nginx-web-app \
    --image-
ids imageDigest=sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2
 \
    --region us-east-1
```

Sortie :

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
```

```
            "imageTag": "temp2.0"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Suppression d'une image dans un référentiel public dans le manuel Amazon ECR Public User Guide.

- Pour plus de détails sur l'API, reportez-vous BatchDeleteImageà la section Référence des AWS CLI commandes.

`create-repository`

L'exemple de code suivant montre comment utiliser`create-repository`.

AWS CLI

Exemple 1 : pour créer un dépôt dans un registre public

L'`create-repository`exemple suivant crée un dépôt nommé project-a/ nginx-web-app dans un registre public.

```
aws ecr-public create-repository \
    --repository-name project-a/nginx-web-app
```

Sortie :

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:08:55.131000+00:00"
    },
    "catalogData": {}
}
```

Pour plus d'informations, consultez la section Création d'un référentiel public dans le manuel Amazon ECR Public User Guide.

Exemple 2 : créer un référentiel dans un registre public avec une brève description du contenu du référentiel, du système et de l'architecture d'exploitation avec lesquels les images du référentiel sont compatibles

L'`create-repository`exemple suivant crée un référentiel nommé project-a/ nginx-web-app dans un registre public avec une brève description du contenu du référentiel, du système et de l'architecture d'exploitation avec lesquels les images du référentiel sont compatibles.

```
aws ecr-public create-repository \
    --repository-name project-a/nginx-web-app \
    --catalog-data 'description=My project-a ECR Public
 Repository,architectures=ARM,ARM 64,x86,x86-64,operatingSystems=Linux'
```

Sortie :

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:23:20.455000+00:00"
    },
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'un référentiel public dans le manuel Amazon ECR Public User Guide.

Exemple 3 : pour créer un référentiel dans un registre public, avec des informations sur AboutText logoImageBlob, UsageText et les balises

L'`create-repository`exemple suivant crée un référentiel nommé project-a/ nginx-web-app dans un registre public, avec les informations AboutText, logoImageBlob UsageText et les balises.

```
aws ecr-public create-repository \
    --cli-input-json file://myfile.json
```

Contenu de `myfile.json` :

```
{
    "repositoryName": "project-a/nginx-web-app",
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",
        "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
 Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
 [`2.0.20200722.0`, `2`, `latest`](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`2.0.20200722.0-
with-sources`, `2-with-sources`, `with-sources`](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n* [`2018.03.0.20200602.1`, `2018.03`, `1`](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n* [`2018.03.0.20200602.1-with-sources`, `2018.03-with-sources`,
 `1-with-sources`](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
 Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
```

```
 to provide a stable, secure, and high-performance execution environment for
 applications running on Amazon EC2. The full distribution includes packages that
 enable easy integration with AWS, including launch configuration tools and many
 popular AWS libraries and tools. AWS provides ongoing security and maintenance
 updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
 contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
 provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
\n\nFor information on security updates for Amazon Linux, please refer to [Amazon
 Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
 Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
 Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
 which does not reflect the state of backported patches for vulnerabilities.\n
\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
 container images in any Docker based environment. Examples include, your laptop,
 in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
 available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
 are available under their own licenses; run `rpm -qi [package name]` or check
 `/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for
 details.\n\nAs with all Docker images, these likely also contain other software
 which may be under other licenses (such as Bash, etc from the base distribution,
 along with any direct or indirect dependencies of the primary software being
 contained).\n\nSome additional license information which was able to be auto-
detected might be found in [the `repo-info` repository's `amazonlinux/` directory]
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
 Security\n\nFor information on security updates for Amazon Linux, please refer
 to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
 and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
 that Docker Hub's vulnerability scanning for Amazon Linux is currently based
 on RPM versions, which does not reflect the state of backported patches for
 vulnerabilities.",
       "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
 can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
 images in any Docker based environment. Examples include, your laptop, in Amazon
 EC2 instances, and ECS clusters.\n\n## How do I install a software package from
 Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
 `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
 Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
 get ongoing updates from Amazon in the form of security updates, bug fix updates,
 and other enhancements. Security bulletins for Amazon Linux are available at
```

```
 https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
 Linux going forward?\n\nYes; in order to avoid any disruption to your existing
 applications and to facilitate migration to Amazon Linux 2, AWS will provide
 regular security updates for Amazon Linux 2018.03 AMI and container image for 2
 years after the final LTS build is announced. You can also use all your existing
 support channels such as AWS Support and Amazon Linux Discussion Forum to continue
 to submit support requests."
    },
    "tags": [
        {
            "Key": "Name",
            "Value": "project-a/nginx-web-app"
        },
        {
            "Key": "Environment",
            "Value": "Prod"
        }
    ]
}
```

Sortie :

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:53:05.749000+00:00"
    },
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
```

```
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/
logo/23861450-4b9b-403c-9a4c-7aa0ef140bb8/2f9bf5a7-a32f-45b4-b5cd-c5770a35e6d7.png",
        "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
 Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
 [`2.0.20200722.0`, `2`, `latest`](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`2.0.20200722.0-
with-sources`, `2-with-sources`, `with-sources`](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n* [`2018.03.0.20200602.1`, `2018.03`, `1`](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n* [`2018.03.0.20200602.1-with-sources`, `2018.03-with-sources`,
 `1-with-sources`](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
 Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
 to provide a stable, secure, and high-performance execution environment for
 applications running on Amazon EC2. The full distribution includes packages that
 enable easy integration with AWS, including launch configuration tools and many
 popular AWS libraries and tools. AWS provides ongoing security and maintenance
 updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
 contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
 provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
\n\nFor information on security updates for Amazon Linux, please refer to [Amazon
 Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
 Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
 Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
 which does not reflect the state of backported patches for vulnerabilities.\n
\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
 container images in any Docker based environment. Examples include, your laptop,
 in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
 available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
 are available under their own licenses; run `rpm -qi [package name]` or check
 `/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for
 details.\n\nAs with all Docker images, these likely also contain other software
 which may be under other licenses (such as Bash, etc from the base distribution,
 along with any direct or indirect dependencies of the primary software being
 contained).\n\nSome additional license information which was able to be auto-
detected might be found in [the `repo-info` repository's `amazonlinux/` directory]
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
```

```
 Security\n\nFor information on security updates for Amazon Linux, please refer
 to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
 and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
 that Docker Hub's vulnerability scanning for Amazon Linux is currently based
 on RPM versions, which does not reflect the state of backported patches for
 vulnerabilities.",
        "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
 can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
 images in any Docker based environment. Examples include, your laptop, in Amazon
 EC2 instances, and ECS clusters.\n\n## How do I install a software package from
 Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
 `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
 Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
 get ongoing updates from Amazon in the form of security updates, bug fix updates,
 and other enhancements. Security bulletins for Amazon Linux are available at
 https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
 Linux going forward?\n\nYes; in order to avoid any disruption to your existing
 applications and to facilitate migration to Amazon Linux 2, AWS will provide
 regular security updates for Amazon Linux 2018.03 AMI and container image for 2
 years after the final LTS build is announced. You can also use all your existing
 support channels such as AWS Support and Amazon Linux Discussion Forum to continue
 to submit support requests."
    }
}
```

Pour plus d'informations, consultez la section Création d'un référentiel public dans le guide de l'utilisateur public Amazon ECR et les données du catalogue du référentiel dans le guide de l'utilisateur public d'Amazon ECR.

- Pour plus de détails sur l'API, reportez-vous CreateRepositoryà la section Référence des AWS CLI commandes.

`delete-repository-policy`

L'exemple de code suivant montre comment utiliser`delete-repository-policy`.

AWS CLI

Pour supprimer une politique de référentiel dans un registre public

L'`delete-repository-policy`exemple suivant montre comment supprimer la politique de dépôt pour le référentiel public ECR de votre AWS compte.

```
aws ecr-public delete-repository-policy \
     --repository-name project-a/nginx-web-app \
     --region us-east-1
```

Sortie :

```
{
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : [ \"arn:aws:iam:"123456789012":user/eksuser1\",
 \"arn:aws:iam:"123456789012":user/admin\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

Pour plus d'informations, consultez la section Suppression d'une déclaration de politique relative à un référentiel public dans le manuel Amazon ECR Public User Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteRepositoryPolicyà la section Référence des AWS CLI commandes.

`delete-repository`

L'exemple de code suivant montre comment utiliser`delete-repository`.

AWS CLI

Pour supprimer un dépôt dans un registre public

L'`delete-repository`exemple suivant supprime un dépôt nommé `project-a/nginx-web-app` dans votre registre public.

```
aws ecr-public delete-repository \
    --repository-name project-a/nginx-web-app
```

Sortie :

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T22:14:50.103000+00:00"
    }
}
```

Pour plus d'informations, consultez Supprimer un référentiel public dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous DeleteRepositoryà la section Référence des AWS CLI commandes.

`describe-image-tags`

L'exemple de code suivant montre comment utiliser`describe-image-tags`.

AWS CLI

Exemple 1 : pour décrire les détails des balises d'image dans le référentiel public

L'`describe-image-tags`exemple suivant décrit les imagetags dans le référentiel `project-a/nginx-web-app` d'exemples.

```
aws ecr-public describe-image-tags \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

Sortie :

```
{
    "imageTagDetails": [
        {
            "imageTag": "latest",
            "createdAt": "2024-07-10T22:29:00-05:00",
            "imageDetail": {
```

```
                "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
                "imageSizeInBytes": 121956548,
                "imagePushedAt": "2024-07-10T22:29:00-05:00",
                "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
                "artifactMediaType": "application/
vnd.docker.container.image.v1+json"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeImageTagsà la section Référence des AWS CLI commandes.

`describe-images`

L'exemple de code suivant montre comment utiliser`describe-images`.

AWS CLI

Exemple 1 : pour décrire des images dans un référentiel de registre public

L'`describe-images`exemple suivant décrit ImagesDetails dans un référentiel nommé `project-a/nginx-web-app` dans un registre public.

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

Sortie :

```
{
    "imageDetails": [
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:0d8c93e72e82fa070d49565c00af32abbe8ddfd7f75e39f4306771ae0628c7e8",
            "imageTags": [
                "temp1.0"
```

```
            ],
            "imageSizeInBytes": 123184716,
            "imagePushedAt": "2024-07-23T11:32:49-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        },
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
            "imageTags": [
                "temp2.0"
            ],
            "imageSizeInBytes": 121956548,
            "imagePushedAt": "2024-07-23T11:39:38-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        },
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
            "imageTags": [
                "temp3.0",
                "latest"
            ],
            "imageSizeInBytes": 232108879,
            "imagePushedAt": "2024-07-22T00:54:34-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        }
    ]
}
```

Pour plus d'informations, consultez Décrire une image dans un référentiel public dans Amazon ECR Public.

Exemple 2 : Pour décrire les images du référentiel en triant ImageTags & imagePushedAt

L'`describe-images`exemple suivant décrit les images d'un dépôt nommé project-a/ nginx-web-app dans un registre public.

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app \
    --query 'sort_by(imageDetails,& imagePushedAt)[*].imageTags[*]' \
    --output text
```

Sortie :

```
temp3.0 latest
temp1.0
temp2.0
```

Exemple 3 : pour décrire les images du référentiel afin de générer les 2 dernières balises d'image insérées dans le référentiel

L'`describe-images`exemple suivant obtient les détails des imagetags à partir du référentiel nommé `project-a/nginx-web-app` dans un registre public et interroge le résultat pour n'afficher que les deux premiers enregistrements.

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app  \
    --query 'sort_by(imageDetails,& imagePushedAt)[*].imageTags[*] | [0:2]' \
    --output text
```

Sortie :

```
temp3.0 latest
temp1.0
```

- Pour plus de détails sur l'API, reportez-vous DescribeImagesà la section Référence des AWS CLI commandes.

`describe-registries`

L'exemple de code suivant montre comment utiliser`describe-registries`.

AWS CLI

Décrire tous les registres d'un registre public

L'`describe-registries`exemple suivant décrit tous les registres de votre compte.

```
aws ecr-public describe-registries
```

Sortie :

```
{
 "registries": [
     {
         "registryId": "123456789012",
         "registryArn": "arn:aws:ecr-public::123456789012:registry/123456789012",
         "registryUri": "public.ecr.aws/publicregistrycustomalias",
         "verified": false,
         "aliases": [
             {
                 "name": "publicregistrycustomalias",
                 "status": "ACTIVE",
                 "primaryRegistryAlias": true,
                 "defaultRegistryAlias": true
             }
         ]
     }
     ]
 }
```

- Pour plus de détails sur l'API, reportez-vous DescribeRegistriesà la section Référence des AWS CLI commandes.

`describe-repository`

L'exemple de code suivant montre comment utiliser`describe-repository`.

AWS CLI

Exemple 1 : pour décrire un dépôt dans un registre public

L'`describe-repositories`exemple suivant décrit un référentiel nommé `project-a/nginx-web-app` dans un registre public.

```
aws ecr-public describe-repositories \
    --repository-name project-a/nginx-web-app
```

Sortie :

```
{
    "repositories": [
        {
            "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
            "createdAt": "2024-07-07T00:07:56.526000-05:00"
        }
    ]
}
```

Exemple 2 : pour décrire tous les référentiels d'un registre public dans un tableau

L'`describe-repositories`exemple suivant décrit tous les référentiels d'un registre public, puis affiche les noms des référentiels sous forme de tableau.

```
aws ecr-public describe-repositories \
    --region us-east-1 \
    --output table \
    --query "repositories[*].repositoryName"
```

Sortie :

```
----------------------------
|   DescribeRepositories    |
+---------------------------+
|  project-a/nginx-web-app  |
|  nginx                    |
|  myfirstrepo1             |
|  helm-test-chart          |
|  test-ecr-public          |
|  nginx-web-app            |
|  sample-repo              |
+---------------------------+
```

- Pour plus de détails sur l'API, reportez-vous DescribeRepositoryà la section Référence des AWS CLI commandes.

`get-authorization-token`

L'exemple de code suivant montre comment utiliser`get-authorization-token`.

AWS CLI

Exemple 1 : pour récupérer un jeton d'autorisation pour tout registre public Amazon ECR auquel le principal IAM a accès

L'`get-authorization-token`exemple suivant obtient un jeton d'autorisation avec la AWS CLI et le définit comme une variable d'environnement.

```
aws ecr-public get-authorization-token \
    --region us-east-1
```

Sortie :

```
{
    "authorizationData": {
        "authorizationToken":
 "QVdTOmV5SndZWGxzYjJKJFHDSFKJHERWUY65IOU36TRYEGFNSDLRIUOTUYTHJKLDFGOcmFUQk9OSFV2UVV4a0x6Sm1
        "expiresAt": "2024-07-25T21:37:26.301000-04:00"
    }
}
```

Pour plus d'informations, consultez les registres publics Amazon ECR dans Amazon ECR Public.

Exemple 2 : pour récupérer un jeton d'autorisation pour tout registre public Amazon ECR auquel le principal IAM a accès

L'`get-authorization-token`exemple suivant obtient un jeton d'autorisation avec la AWS CLI et le définit comme une variable d'environnement.

```
aws ecr-public get-authorization-token \
    --region us-east-1 \
    --output=text \
    --query 'authorizationData.authorizationToken'
```

Sortie :

```
QVdTOmV5SndZWGxzYjJKJFHDSFKJHERWUY65IOU36TRYEGFNSDLRIUOTUYTHJKLDFGOcmFUQk9OSFV2UVV4a0x6Sm1ZV
```

Pour plus d'informations, consultez les registres publics Amazon ECR dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous GetAuthorizationTokenà la section Référence des AWS CLI commandes.

`get-login-password`

L'exemple de code suivant montre comment utiliser`get-login-password`.

AWS CLI

Exemple 1 : pour authentifier le docker auprès d'un registre public Amazon ECR

L'`get-login-password`exemple suivant récupère et affiche un jeton d'authentification à l'aide de l' GetAuthorizationToken API que vous pouvez utiliser pour vous authentifier auprès d'un registre public Amazon ECR.

```
aws ecr-public get-login-password \
    --region us-
east-1
| docker login \
    --username AWS \
    --password-stdin public.ecr.aws
```

Cette commande ne produit aucune sortie dans le terminal mais dirige la sortie vers Docker.

Pour plus d'informations, consultez Authentifier auprès du registre public dans Amazon ECR Public.

Exemple 2 : pour authentifier docker auprès de votre propre registre public AmazonECR personnalisé

L'`get-login-password`exemple suivant récupère et affiche un jeton d'authentification à l'aide de l' GetAuthorizationToken API que vous pouvez utiliser pour vous authentifier auprès de votre propre registre public Amazon ECR personnalisé.

```
 aws ecr-public get-login-password \
    --region us-east-1 \
| docker login \
    --username AWS \
    --password-stdin public.ecr.aws/<your-public-registry-custom-alias>
```

Cette commande ne produit aucune sortie dans le terminal mais dirige la sortie vers Docker.

Pour plus d'informations, consultez Authentifiez-vous auprès de votre propre Amazon ECR Public dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous GetLoginPasswordà la section Référence des AWS CLI commandes.

`get-registry-catalog-data`

L'exemple de code suivant montre comment utiliser`get-registry-catalog-data`.

AWS CLI

Pour récupérer les métadonnées du catalogue pour un registre ECR public

Ce qui suit `get-registry-catalog-data` permet de récupérer les métadonnées du catalogue pour un registre public ECR.

```
aws ecr-public get-registry-catalog-data \
    --region us-east-1
```

Sortie :

```
{
    "registryCatalogData": {
        "displayName": "YourCustomPublicRepositoryalias"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetRegistryCatalogDataà la section Référence des AWS CLI commandes.

`get-repository-catalog-data`

L'exemple de code suivant montre comment utiliser`get-repository-catalog-data`.

AWS CLI

Pour récupérer les métadonnées d'un catalogue pour un référentiel dans un registre public

L'`get-repository-catalog-data`exemple suivant répertorie les métadonnées du catalogue pour le référentiel `project-a/nginx-web-app` dans un registre public.

```
aws ecr-public get-repository-catalog-data \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

Sortie :

```
{
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/logo/491d3846-8f33-4d8b-
a10c-c2ce271e6c0d/4f09d87c-2569-4916-a932-5c296bf6f88a.png",
        "aboutText": "## Quick reference\n\nMaintained <truncated>",
        "usageText": "## Supported architectures\n\namd64, arm64v8\n\n##
 <truncated>"
    }
}
```

Pour plus d'informations, consultez les données du catalogue du référentiel dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous GetRepositoryCatalogDataà la section Référence des AWS CLI commandes.

`get-repository-policy`

L'exemple de code suivant montre comment utiliser`get-repository-policy`.

AWS CLI

Pour obtenir une politique de dépôt associée au référentiel

L'`get-repository-policy`exemple suivant permet d'obtenir une politique de dépôt associée au dépôt.

```
aws ecr-public get-repository-policy \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

Sortie :

```
{
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : [ \"arn:aws:iam::123456789012:user/eksuser1\",
 \"arn:aws:iam::123456789012:user/admin\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

Pour plus d'informations, consultez la section Utilisation GetRepositoryPolicy avec un AWS SDK ou une CLI dans le manuel Amazon ECR Public User Guide.

- Pour plus de détails sur l'API, reportez-vous GetRepositoryPolicyà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un dépôt public dans un registre public

L'`list-tags-for-resource`exemple suivant répertorie les balises d'une ressource nommée `project-a/nginx-web-app` dans un registre public.

```
aws ecr-public list-tags-for-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-
app \
```

```
    --region us-east-1
```

Sortie :

```
{
    "tags": [
        {
            "Key": "Environment",
            "Value": "Prod"
        },
        {
            "Key": "stack",
            "Value": "dev1"
        },
        {
            "Key": "Name",
            "Value": "project-a/nginx-web-app"
        }
    ]
}
```

Pour plus d'informations, consultez Lister les balises d'un référentiel public dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-registry-catalog-data`

L'exemple de code suivant montre comment utiliser`put-registry-catalog-data`.

AWS CLI

Pour créer ou mettre à jour les métadonnées du catalogue pour un registre ECR public

Ce qui suit `put-registry-catalog-data` crée ou met à jour les métadonnées du catalogue pour un registre public ECR. Seuls les comptes dotés du badge de compte vérifié peuvent avoir un nom d'affichage dans le registre.

```
aws ecr-public put-registry-catalog-data \
    --region us-east-1 \
    --display-name <YourCustomPublicRepositoryalias>
```

Sortie :

```
{
    "registryCatalogData": {
        "displayName": "YourCustomPublicRepositoryalias"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutRegistryCatalogDataà la section Référence des AWS CLI commandes.

`put-repository-catalog-data`

L'exemple de code suivant montre comment utiliser`put-repository-catalog-data`.

AWS CLI

Pour créer ou mettre à jour les données du catalogue pour un référentiel dans un registre public

L'`put-repository-catalog-data`exemple suivant crée ou met à jour des données de catalogue pour le référentiel nommé project-a/ nginx-web-app dans un registre public, ainsi que des informations sur AboutText, UsageText et les logoImageBlob balises.

```
aws ecr-public put-repository-catalog-data \
    --repository-name project-a/nginx-web-app \
    --cli-input-json file://repository-catalog-data.json \
    --region us-east-1
```

Contenu de `repository-catalog-data.json` :

```
{
    "repositoryName": "project-a/nginx-web-app",
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
```

```
            "Linux"
        ],
        "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",
        "aboutText": "## Quick reference.",
        "usageText": "## Supported architectures are as follows."
    }
}
```

Sortie :

```
{
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/logo/df86cf58-ee60-4061-
b804-0be24d97ccb1/4a9ed9b2-69e4-4ede-b924-461462d20ef0.png",
        "aboutText": "## Quick reference.",
        "usageText": "## Supported architectures are as follows."
    }
}
```

Pour plus d'informations, consultez les données du catalogue du référentiel dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous PutRepositoryCatalogDataà la section Référence des AWS CLI commandes.

`set-repository-policy`

L'exemple de code suivant montre comment utiliser`set-repository-policy`.

AWS CLI

Exemple 1 : pour définir une politique de dépôt afin d'autoriser une extraction sur le référentiel

L'`set-repository-policy`exemple suivant applique une politique de référentiel public ECR au référentiel spécifié afin de contrôler les autorisations d'accès.

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

Contenu de `my-repository-policy.json` :

```
{
    "Version" : "2008-10-17",
    "Statement" : [
        {
            "Sid" : "allow public pull",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

Sortie :

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

Pour plus d'informations, consultez la section Définition d'une déclaration de politique de dépôt dans le manuel Amazon ECR Public User Guide.

Exemple 2 : pour définir une politique de référentiel afin d'autoriser un utilisateur IAM de votre compte à envoyer des images

L'`set-repository-policy`exemple suivant permet à un utilisateur IAM de votre compte de transférer des images vers un référentiel ECR de votre AWS compte en utilisant le fichier d'entrée `file://my-repository-policy.json` nommé texte de politique.

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

Contenu de `my-repository-policy.json` :

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowPush",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-id:user/push-pull-user-1",
                    "arn:aws:iam::account-id:user/push-pull-user-2"
                ]
            },
            "Action": [
                "ecr-public:BatchCheckLayerAvailability",
                "ecr-public:PutImage",
                "ecr-public:InitiateLayerUpload",
                "ecr-public:UploadLayerPart",
                "ecr-public:CompleteLayerUpload"
            ]
        }
    ]
}
```

Sortie :

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" :
 [ {\n    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n      \"AWS\" : [ \"arn:aws:iam::12345678901:user/admin
\", \"arn:aws:iam::12345678901:user/eksuser1\" ]\n    },\n    \"Action\" :
```

```
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

Pour plus d'informations, consultez la section Définition d'une déclaration de politique de dépôt dans le manuel Amazon ECR Public User Guide.

Exemple 3 : pour définir une politique de référentiel afin d'autoriser un utilisateur IAM d'un autre compte à envoyer des images

L'`set-repository-policy`exemple suivant permet à un compte spécifique d'envoyer des images en utilisant cli input file ://my-repository-policy.json dans votre AWS compte.

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

Contenu de `my-repository-policy.json` :

```
 {
     "Version": "2008-10-17",
     "Statement": [
         {
             "Sid": "AllowCrossAccountPush",
             "Effect": "Allow",
             "Principal": {
                 "AWS": "arn:aws:iam::other-or-same-account-id:role/RoleName"
             },
             "Action": [
                 "ecr-public:BatchCheckLayerAvailability",
                 "ecr-public:PutImage",
                 "ecr-public:InitiateLayerUpload",
                 "ecr-public:UploadLayerPart",
                 "ecr-public:CompleteLayerUpload"
             ]
         }
     ]
}
```

Sortie :

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"AllowCrossAccountPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : \"arn:aws:iam::12345678901:role/RoleName\"\n    },\n
 \"Action\" : [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage
\", \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

Pour plus d'informations, consultez les exemples de politiques relatives aux référentiels publics dans le manuel Amazon ECR Public User Guide.

- Pour plus de détails sur l'API, reportez-vous SetRepositoryPolicyà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Exemple 1 : Pour baliser un dépôt public existant dans un registre public

L'`tag-resource`exemple suivant balise un référentiel nommé `project-a/nginx-web-app` dans un registre public.

```
aws ecr-public tag-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-
app \
    --tags Key=stack,Value=dev \
    --region us-east-1
```

Pour plus d'informations, consultez Utilisation de balises pour un référentiel public dans Amazon ECR Public.

Exemple 2 : Pour baliser un référentiel public existant avec plusieurs balises dans un registre public.

L'`tag-resource`exemple suivant ajoute plusieurs balises à un référentiel existant.

```
aws ecr-public tag-resource \
    --resource-arn arn:aws:ecr-public::890517186334:repository/project-a/nginx-web-app  \
    --tags Key=key1,Value=value1 Key=key2,Value=value2 Key=key3,Value=value3 \
    --region us-east-1
```

Pour plus d'informations, consultez Utilisation de balises pour un référentiel public dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Exemple 1 : pour débaliser un dépôt public existant dans un registre public

L'`untag-resource`exemple suivant balise un référentiel nommé `project-a/nginx-web-app` dans un registre public.

```
aws ecr-public untag-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-app \
    --tag-keys stack \
    --region us-east-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Utilisation de balises pour un référentiel public dans Amazon ECR Public.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples d'Amazon ECS utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon ECS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`capacity-provider-update`

L'exemple de code suivant montre comment utiliser`capacity-provider-update`.

AWS CLI

Mettre à jour le fournisseur de capacité dans un cluster ECS

L'`update-capacity-provider`exemple suivant montre comment modifier les paramètres du fournisseur de capacité dans un cluster ECS.

```
aws ecs update-capacity-provider \
    --name Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt \
    --auto-scaling-group-
provider "managedScaling={status=DISABLED,targetCapacity=50,minimumScalingStepSize=2,maximum
```

Sortie :

```
{
    "capacityProvider": {
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt",
        "name": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:424941d1-b43f-4a17-
```

```
adbb-08b6a6e397e1:autoScalingGroupName/Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-ECSAutoScalingGroup-f44jrQHS2nRB",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 10000,
                "instanceWarmupPeriod": 300
            },
            "managedTerminationProtection": "DISABLED",
            "managedDraining": "ENABLED"
        },
        "updateStatus": "UPDATE_IN_PROGRESS",
        "tags": []
    }
}
```

Pour plus d'informations sur le fournisseur de capacité, consultez les fournisseurs de capacité Amazon ECS pour le type de EC2 lancement dans le guide du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous CapacityProviderUpdateà la section Référence des AWS CLI commandes.

`create-capacity-provider`

L'exemple de code suivant montre comment utiliser`create-capacity-provider`.

AWS CLI

Pour créer un fournisseur de capacité

L' create-capacity-providerexemple suivant crée un fournisseur de capacité qui utilise un groupe Auto Scaling nommé MyASG, dont le dimensionnement géré est activé et la protection contre les interruptions de service activée. Cette configuration est utilisée pour la scalabilité automatique de cluster Amazon ECS.

```
aws ecs create-capacity-provider \
    --name "MyCapacityProvider" \
    --auto-scaling-group-provider "autoScalingGroupArn=arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/
MyASG,managedScaling={status=ENABLED,targetCapacity=100},managedTerminationProtection=ENABLE
```

Sortie :

```
{
    "capacityProvider": {
    "capacityProviderArn": "arn:aws:ecs:us-east-1:123456789012:capacity-provider/
MyCapacityProvider",
    "name": "MyCapacityProvider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
        "autoScalingGroupArn": "arn:aws:autoscaling:us-
east-1:132456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/MyASG",
        "managedScaling": {
            "status": "ENABLED",
            "targetCapacity": 100,
            "minimumScalingStepSize": 1,
            "maximumScalingStepSize": 10000,
            "instanceWarmupPeriod": 300
        },
        "managedTerminationProtection": "ENABLED"
    },
    "tags": []
}
```

Pour plus d'informations, consultez la section Amazon ECS Cluster Auto Scaling dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous CreateCapacityProviderà la section Référence des AWS CLI commandes.

`create-cluster`

L'exemple de code suivant montre comment utiliser`create-cluster`.

AWS CLI

Exemple 1 : pour créer un nouveau cluster

L'`create-cluster`exemple suivant crée un cluster nommé `MyCluster` et active CloudWatch Container Insights avec une observabilité améliorée.

```
aws ecs create-cluster \
```

```
    --cluster-name MyCluster \
    --settings name=containerInsights,value=enhanced
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enhanced"
            }
        ],
        "tags": []
    }
}
```

Pour plus d'informations, consultez la section Création d'un cluster dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour créer un nouveau cluster à l'aide de fournisseurs de capacité

L'`create-cluster`exemple suivant crée un cluster et y associe deux fournisseurs de capacité existants. La `create-capacity-provider` commande est utilisée pour créer un fournisseur de capacité. La spécification d'une stratégie de fournisseur de capacité par défaut est facultative, mais recommandée. Dans cet exemple, nous créons un cluster nommé `MyCluster` et y associons les `MyCapacityProvider1` fournisseurs de `MyCapacityProvider2` capacité. Une stratégie de fournisseur de capacité par défaut est spécifiée pour répartir les tâches de manière égale entre les deux fournisseurs de capacité.

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
```

```
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "PROVISIONING",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1",
            "MyCapacityProvider2"
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "MyCapacityProvider1",
                "weight": 1,
                "base": 0
            },
            {
                "capacityProvider": "MyCapacityProvider2",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
           {
               "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
               "type": "asp",
               "status": "PRECREATED",
               "details": [
```

```
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
                    {
                        "name": "scalingPlanName",
                        "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "asp",
                "status": "PRECREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPlanName",
                        "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

Exemple 3 : pour créer un nouveau cluster avec plusieurs balises

L'`create-cluster`exemple suivant crée un cluster avec plusieurs balises. Pour plus d'informations sur l'ajout de balises à l'aide d'une syntaxe abrégée, consultez la section Utilisation de la syntaxe abrégée avec l'interface de ligne de AWS commande dans le guide de l'utilisateur de la CLI AWS .

```
aws ecs create-cluster \
```

```
    --cluster-name MyCluster \
    --tags key=key1,value=value1 key=key2,value=value2
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [
            {
                "key": "key1",
                "value": "value1"
            },
            {
                "key": "key2",
                "value": "value2"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'un cluster dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateClusterà la section Référence des AWS CLI commandes.

`create-service`

L'exemple de code suivant montre comment utiliser`create-service`.

AWS CLI

Exemple 1 : pour créer un service avec une tâche Fargate

L'`create-service`exemple suivant montre comment créer un service à l'aide d'une tâche Fargate.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name MyService \
    --task-definition sample-fargate:1 \
    --desired-count 2 \
    --launch-type FARGATE \
    --platform-version LATEST \
    --network-configuration
 'awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp
 \
    --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/
MyService",
        "serviceName": "MyService",
          "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:1",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/1234567890123456789",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sample-fargate:1",
```

```
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1557119253.821,
                "updatedAt": 1557119253.821,
                "launchType": "FARGATE",
                "platformVersion": "1.3.0",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                }
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": 1557119253.821,
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "schedulingStrategy": "REPLICA",
        "tags": [
            {
                "key": "key1",
                "value": "value1"
            },
            {
```

```
                "key": "key2",
                "value": "value2"
            },
            {
                "key": "key3",
                "value": "value3"
            }
        ],
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

Pour plus d'informations, consultez la section Création d'un service dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour créer un service à l'aide du type de EC2 lancement

L'`create-service`exemple suivant montre comment créer un service appelé `ecs-simple-service` avec une tâche utilisant le type de EC2 lancement. Le service utilise la définition de `sleep360` tâche et conserve une seule instanciation de la tâche.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service \
    --task-definition sleep360:2 \
    --desired-count 1
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/ecs-
simple-service",
        "serviceName": "ecs-simple-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
```

```
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sleep360:2",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/1234567890123456789",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:2",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1557206498.798,
                "updatedAt": 1557206498.798,
                "launchType": "EC2"
            }
        ],
        "events": [],
        "createdAt": 1557206498.798,
        "placementConstraints": [],
        "placementStrategy": [],
        "schedulingStrategy": "REPLICA",
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

Pour plus d'informations, consultez la section Création d'un service dans le manuel Amazon ECS Developer Guide.

Exemple 3 : pour créer un service utilisant un contrôleur de déploiement externe

L'`create-service`exemple suivant crée un service qui utilise un contrôleur de déploiement externe.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name MyService \
    --deployment-controller type=EXTERNAL \
    --desired-count 1
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/
MyService",
        "serviceName": "MyService",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "taskSets": [],
        "deployments": [],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": 1557128207.101,
        "placementConstraints": [],
        "placementStrategy": [],
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "EXTERNAL"
        },
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

Pour plus d'informations, consultez la section Création d'un service dans le manuel Amazon ECS Developer Guide.

Exemple 4 : pour créer un nouveau service derrière un équilibreur de charge

L'`create-service`exemple suivant montre comment créer un service situé derrière un équilibreur de charge. Un équilibreur de charge doit être configuré dans la même région que votre

instance de conteneur. Cet exemple utilise l'`--cli-input-json`option et un fichier d'entrée JSON appelé `ecs-simple-service-elb.json` avec le contenu suivant.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service-elb \
    --cli-input-json file://ecs-simple-service-elb.json
```

Contenu de `ecs-simple-service-elb.json` :

```
 {
    "serviceName": "ecs-simple-service-elb",
    "taskDefinition": "ecs-demo",
    "loadBalancers": [
        {
            "loadBalancerName": "EC2Contai-EcsElast-123456789012",
            "containerName": "simple-demo",
            "containerPort": 80
        }
    ],
    "desiredCount": 10,
    "role": "ecsServiceRole"
}
```

Sortie :

```
{
    "service": {
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/ecs-
demo:1",
        "pendingCount": 0,
        "loadBalancers": [
            {
                "containerName": "ecs-demo",
                "containerPort": 80,
                "loadBalancerName": "EC2Contai-EcsElast-123456789012"
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/ecsServiceRole",
        "desiredCount": 10,
        "serviceName": "ecs-simple-service-elb",
```

```
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/ecs-simple-
service-elb",
        "deployments": [
            {
                "status": "PRIMARY",
                "pendingCount": 0,
                "createdAt": 1428100239.123,
                "desiredCount": 10,
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/ecs-demo:1",
                "updatedAt": 1428100239.123,
                "id": "ecs-svc/1234567890123456789",
                "runningCount": 0
            }
        ],
        "events": [],
        "runningCount": 0
    }
}
```

Pour plus d'informations, consultez Utiliser l'équilibrage de charge pour distribuer le trafic des services Amazon ECS dans le manuel Amazon ECS Developer Guide.

Exemple 5 : pour configurer les volumes Amazon EBS lors de la création du service

L'`create-service`exemple suivant montre comment configurer les volumes Amazon EBS pour chaque tâche gérée par le service. Vous devez avoir un rôle d'infrastructure Amazon ECS configuré avec la politique `AmazonECSInfrastructureRolePolicyForVolumes` gérée attachée. Vous devez spécifier une définition de tâche portant le même nom de volume que dans la `create-service` demande. Cet exemple utilise l'`--cli-input-json`option et un fichier d'entrée JSON appelé `ecs-simple-service-ebs.json` avec le contenu suivant.

```
aws ecs create-service \
    --cli-input-json file://ecs-simple-service-ebs.json
```

Contenu de `ecs-simple-service-ebs.json` :

```
{
    "cluster": "mycluster",
    "taskDefinition": "mytaskdef",
    "serviceName": "ecs-simple-service-ebs",
```

```
    "desiredCount": 2,
    "launchType": "FARGATE",
    "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
    "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
            }
        }
   ]
}
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/ecs-
simple-service-ebs",
        "serviceName": "ecs-simple-service-ebs",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:3",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
```

```
                "enable": false,
                "rollback": false
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/7851020056849183687",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:3",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-01-21T11:32:38.034000-06:00",
                "updatedAt": "2025-01-21T11:32:38.034000-06:00",
                "launchType": "EC2",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "DISABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7851020056849183687 in
 progress.",
                "volumeConfigurations": [
                    {
                        "name": "myEBSVolume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
```

```
                        }
                    }
                ]
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": "2025-01-21T11:32:38.034000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:user/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": false,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "DISABLED"
    }
}
```

Pour plus d'informations, consultez la section Utiliser les volumes Amazon EBS avec Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous CreateServiceà la section Référence des AWS CLI commandes.

`create-task-set`

L'exemple de code suivant montre comment utiliser`create-task-set`.

AWS CLI

Pour créer un ensemble de tâches

L'`create-task-set`exemple suivant crée un ensemble de tâches dans un service qui utilise un contrôleur de déploiement externe.

```
aws ecs create-task-set \
    --cluster MyCluster \
    --service MyService \
    --task-definition MyTaskDefinition:2 \
    --network-
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

Sortie :

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
MyTaskDefinition:2",
        "computedDesiredCount": 0,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557128360.711,
        "updatedAt": 1557128360.711,
        "launchType": "EC2",
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
```

```
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 0.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557128360.711
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateTaskSetà la section Référence des AWS CLI commandes.

`delete-account-setting`

L'exemple de code suivant montre comment utiliser`delete-account-setting`.

AWS CLI

Pour supprimer les paramètres du compte pour un utilisateur ou un rôle IAM spécifique

L'exemple suivant `delete-account-setting` supprime les paramètres du compte pour un utilisateur ou un rôle IAM spécifique.

```
aws ecs delete-account-setting \
    --name serviceLongArnFormat \
    --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Sortie :

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

Pour plus d'informations, consultez Amazon Resource Names (ARNs) et IDs le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DeleteAccountSettingà la section Référence des AWS CLI commandes.

`delete-attributes`

L'exemple de code suivant montre comment utiliser`delete-attributes`.

AWS CLI

Pour supprimer un ou plusieurs attributs personnalisés d'une ressource Amazon ECS

Ce qui suit `delete-attributes` supprime un attribut portant le nom `stack` d'une instance de conteneur.

```
aws ecs delete-attributes \
    --attributes name=stack,targetId=arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Sortie :

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAttributesà la section Référence des AWS CLI commandes.

`delete-capacity-provider`

L'exemple de code suivant montre comment utiliser`delete-capacity-provider`.

AWS CLI

Exemple 1 : pour supprimer un fournisseur de capacité à l'aide de l'Amazon Resource Name (ARN)

L'`delete-capacity-provider`exemple suivant supprime un fournisseur de capacité en spécifiant son Amazon Resource Name (ARN). L'ARN ainsi que l'état de suppression du fournisseur de capacité peuvent être récupérés à l'aide de la `describe-capacity-providers` commande.

```
aws ecs delete-capacity-provider \
    --capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/
ExampleCapacityProvider
```

Sortie :

```
{
    "capacityProvider": {
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/ExampleCapacityProvider",
        "name": "ExampleCapacityProvider",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 10000
            },
            "managedTerminationProtection": "DISABLED"
        },
        "updateStatus": "DELETE_IN_PROGRESS",
        "tags": []
    }
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour supprimer un fournisseur de capacité en utilisant le nom

L'`delete-capacity-provider`exemple suivant supprime un fournisseur de capacité en spécifiant son nom abrégé. Le nom abrégé ainsi que l'état de la suppression du fournisseur de capacité peuvent être récupérés à l'aide de la `describe-capacity-providers` commande.

```
aws ecs delete-capacity-provider \
    --capacity-provider ExampleCapacityProvider
```

Sortie :

```
{
    "capacityProvider": {
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/ExampleCapacityProvider",
        "name": "ExampleCapacityProvider",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 10000
            },
            "managedTerminationProtection": "DISABLED"
        },
        "updateStatus": "DELETE_IN_PROGRESS",
        "tags": []
    }
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteCapacityProviderà la section Référence des AWS CLI commandes.

`delete-cluster`

L'exemple de code suivant montre comment utiliser`delete-cluster`.

AWS CLI

Pour supprimer un cluster vide

L'`delete-cluster`exemple suivant supprime le cluster vide spécifié.

```
aws ecs delete-cluster --cluster MyCluster
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "status": "INACTIVE",
        "clusterName": "MyCluster",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0
        "statistics": [],
        "tags": []
    }
}
```

Pour plus d'informations, consultez Supprimer un cluster dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteClusterà la section Référence des AWS CLI commandes.

`delete-service`

L'exemple de code suivant montre comment utiliser`delete-service`.

AWS CLI

Pour supprimer un service

L'`ecs delete-service`exemple suivant supprime le service spécifié d'un cluster. Vous pouvez inclure le `--force` paramètre pour supprimer un service même s'il n'a pas été redimensionné à zéro tâche.

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

Pour plus d'informations, consultez Supprimer un service dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceà la section Référence des AWS CLI commandes.

`delete-task-definitions`

L'exemple de code suivant montre comment utiliser`delete-task-definitions`.

AWS CLI

Pour supprimer une définition de tâche

L'`delete-task-definitions`exemple suivant supprime une définition de tâche INACTIVE.

```
aws ecs delete-task-definitions \
    --task-definition curltest:1
```

Sortie :

```
{
"taskDefinitions": [
    {
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
curltest:1",
        "containerDefinitions": [
            {
                "name": "ctest",
                "image": "mreferre/eksutils",
                "cpu": 0,
                "portMappings": [],
                "essential": true,
                "entryPoint": [
                    "sh",
                    "-c"
```

```
                ],
                "command": [
                    "curl ${ECS_CONTAINER_METADATA_URI_V4}/task"
                ],
                "environment": [],
                "mountPoints": [],
                "volumesFrom": [],
                "logConfiguration": {
                    "logDriver": "awslogs",
                    "options": {
                        "awslogs-create-group": "true",
                        "awslogs-group": "/ecs/curltest",
                        "awslogs-region": "us-east-1",
                        "awslogs-stream-prefix": "ecs"
                    }
                }
            }
        ],
        "family": "curltest",
        "taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
        "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
        "networkMode": "awsvpc",
        "revision": 1,
        "volumes": [],
        "status": "DELETE_IN_PROGRESS",
        "compatibilities": [
            "EC2",
            "FARGATE"
        ],
        "requiresCompatibilities": [
            "FARGATE"
        ],
        "cpu": "256",
        "memory": "512",
        "registeredAt": "2021-09-10T12:56:24.704000+00:00",
        "deregisteredAt": "2023-03-14T15:20:59.419000+00:00",
        "registeredBy": "arn:aws:sts::123456789012:assumed-role/Admin/jdoe"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DeleteTaskDefinitionsà la section Référence des AWS CLI commandes.

`delete-task-set`

L'exemple de code suivant montre comment utiliser`delete-task-set`.

AWS CLI

Pour supprimer un ensemble de tâches

L'`delete-task-set`exemple suivant montre comment supprimer un ensemble de tâches. Vous pouvez inclure le `--force` paramètre pour supprimer un ensemble de tâches même s'il n'a pas été redimensionné à zéro.

```
aws ecs delete-task-set \
    --cluster MyCluster \
    --service MyService \
    --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-
svc/1234567890123456789 \
    --force
```

Sortie :

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "DRAINING",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
        "computedDesiredCount": 0,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557130260.276,
        "updatedAt": 1557130290.707,
        "launchType": "EC2",
        "networkConfiguration": {
```

```
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12345678"
                ],
                "securityGroups": [
                    "sg-12345678"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 0.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557130290.707
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteTaskSetà la section Référence des AWS CLI commandes.

`deregister-container-instance`

L'exemple de code suivant montre comment utiliser`deregister-container-instance`.

AWS CLI

Pour désenregistrer une instance de conteneur d'un cluster

L'`deregister-container-instance`exemple suivant annule l'enregistrement d'une instance de conteneur du cluster spécifié. Si des tâches sont toujours en cours d'exécution dans l'instance de conteneur, vous devez soit arrêter ces tâches avant de vous désinscrire, soit utiliser l'option. `--force`

```
aws ecs deregister-container-instance \
    --cluster arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
    --container-instance arn:aws:ecs:us-west-2:123456789012:container-instance/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --force
```

Sortie :


```
{
    "containerInstance": {
        "remainingResources": [
            {
                "integerValue": 1024,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "CPU"
            },
            {
                "integerValue": 985,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "MEMORY"
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS",
                "stringSetValue": [
                    "22",
                    "2376",
                    "2375",
                    "51678",
                    "51679"
                ],
                "longValue": 0,
                "doubleValue": 0.0
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS_UDP",
                "stringSetValue": [],
                "longValue": 0,
                "doubleValue": 0.0
            }
        ],
        "agentConnected": true,
        "attributes": [
```

```
            {
                "name": "ecs.capability.secrets.asm.environment-variables"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
            },
            {
                "value": "ami-01a82c3fce2c3ba58",
                "name": "ecs.ami-id"
            },
            {
                "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.none"
            },
            {
                "name": "ecs.capability.ecr-endpoint"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
            },
            {
                "value": "vpc-1234567890123467",
                "name": "ecs.vpc-id"
            },
            {
                "name": "ecs.capability.execution-role-awslogs"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
            },
            {
                "name": "ecs.capability.docker-plugin.local"
            },
            {
                "name": "ecs.capability.task-eni"
            },
```

```
            {
                "name": "ecs.capability.task-cpu-mem-limit"
            },
            {
                "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
            },
            {
                "name": "ecs.capability.execution-role-ecr-pull"
            },
            {
                "name": "ecs.capability.container-health-check"
            },
            {
                "value": "subnet-1234567890123467",
                "name": "ecs.subnet-id"
            },
            {
                "value": "us-west-2a",
                "name": "ecs.availability-zone"
            },
            {
                "value": "t2.micro",
                "name": "ecs.instance-type"
            },
            {
                "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
            },
            {
                "name": "ecs.capability.aws-appmesh"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
```

```
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
            },
            {
                "name": "com.amazonaws.ecs.capability.privileged-container"
            },
            {
                "name": "ecs.capability.container-ordering"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
            },
            {
                "value": "x86_64",
                "name": "ecs.cpu-architecture"
            },
            {
                "value": "93f43776-2018.10.0",
                "name": "ecs.capability.cni-plugin-version"
            },
            {
                "name": "ecs.capability.secrets.ssm.environment-variables"
            },
            {
                "name": "ecs.capability.pid-ipc-namespace-sharing"
            },
            {
                "name": "com.amazonaws.ecs.capability.ecr-auth"
            },
            {
                "value": "linux",
                "name": "ecs.os-type"
            },
            {
```

```
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
            },
            {
                "name": "ecs.capability.task-eia"
            },
            {
                "name": "ecs.capability.private-registry-
authentication.secretsmanager"
            },
            {
                "name": "com.amazonaws.ecs.capability.task-iam-role"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
            }
        ],
        "pendingTasksCount": 0,
        "tags": [],
        "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "registeredResources": [
            {
                "integerValue": 1024,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "CPU"
            },
            {
                "integerValue": 985,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "MEMORY"
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
```

```
                "name": "PORTS",
                "stringSetValue": [
                    "22",
                    "2376",
                    "2375",
                    "51678",
                    "51679"
                ],
                "longValue": 0,
                "doubleValue": 0.0
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS_UDP",
                "stringSetValue": [],
                "longValue": 0,
                "doubleValue": 0.0
            }
        ],
        "status": "INACTIVE",
        "registeredAt": 1557768075.681,
        "version": 4,
        "versionInfo": {
            "agentVersion": "1.27.0",
            "agentHash": "aabe65ee",
            "dockerVersion": "DockerVersion: 18.06.1-ce"
        },
        "attachments": [],
        "runningTasksCount": 0,
        "ec2InstanceId": "i-12345678901234678"
    }
}
```

Pour plus d'informations, consultez la section Désenregistrer une instance de conteneur dans le manuel du développeur ECS.

- Pour plus de détails sur l'API, reportez-vous DeregisterContainerInstanceà la section Référence des AWS CLI commandes.

`deregister-task-definition`

L'exemple de code suivant montre comment utiliser`deregister-task-definition`.

AWS CLI

Pour annuler l'enregistrement d'une définition de tâche

L'`deregister-task-definition`exemple suivant annule l'enregistrement de la première révision de la définition de `curler` tâche dans votre région par défaut.

```
aws ecs deregister-task-definition --task-definition curler:1
```

Notez que dans le résultat obtenu, l'état de la définition de la tâche indique `INACTIVE` :

```
{
    "taskDefinition": {
        "status": "INACTIVE",
        "family": "curler",
        "volumes": [],
        "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/
curler:1",
        "containerDefinitions": [
            {
                "environment": [],
                "name": "curler",
                "mountPoints": [],
                "image": "curl:latest",
                "cpu": 100,
                "portMappings": [],
                "entryPoint": [],
                "memory": 256,
                "command": [
                    "curl -v http://example.com/"
                ],
                "essential": true,
                "volumesFrom": []
            }
        ],
        "revision": 1
    }
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DeregisterTaskDefinitionà la section Référence des AWS CLI commandes.

`describe-capacity-providers`

L'exemple de code suivant montre comment utiliser`describe-capacity-providers`.

AWS CLI

Exemple 1 : Décrire tous les fournisseurs de capacité

L'`describe-capacity-providers`exemple suivant permet de récupérer des informations sur tous les fournisseurs de capacité.

```
aws ecs describe-capacity-providers
```

Sortie :

```
{
    "capacityProviders": [
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/MyCapacityProvider",
            "name": "MyCapacityProvider",
            "status": "ACTIVE",
            "autoScalingGroupProvider": {
                "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
                "managedScaling": {
                    "status": "ENABLED",
                    "targetCapacity": 100,
                    "minimumScalingStepSize": 1,
                    "maximumScalingStepSize": 1000
                },
                "managedTerminationProtection": "ENABLED"
            },
            "tags": []
        },
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE",
```

```
            "name": "FARGATE",
            "status": "ACTIVE",
            "tags": []
        },
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE_SPOT",
            "name": "FARGATE_SPOT",
            "status": "ACTIVE",
            "tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

Exemple 2 : Pour décrire un fournisseur de capacité spécifique

L'`describe-capacity-providers`exemple suivant permet de récupérer des informations sur un fournisseur de capacité spécifique. L'utilisation du `--include TAGS` paramètre ajoutera les balises associées au fournisseur de capacité à la sortie.

```
aws ecs describe-capacity-providers \
    --capacity-providers MyCapacityProvider \
    --include TAGS
```

Sortie :

```
{
    "capacityProviders": [
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/MyCapacityProvider",
            "name": "MyCapacityProvider",
            "status": "ACTIVE",
            "autoScalingGroupProvider": {
                "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
                "managedScaling": {
                    "status": "ENABLED",
                    "targetCapacity": 100,
```

```
                    "minimumScalingStepSize": 1,
                    "maximumScalingStepSize": 1000
                },
                "managedTerminationProtection": "ENABLED"
            },
            "tags": [
                {
                    "key": "environment",
                    "value": "production"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeCapacityProvidersà la section Référence des AWS CLI commandes.

`describe-clusters`

L'exemple de code suivant montre comment utiliser`describe-clusters`.

AWS CLI

Exemple 1 : pour décrire un cluster

L'`describe-clusters`exemple suivant permet de récupérer des informations sur le cluster spécifié.

```
aws ecs describe-clusters \
    --cluster default
```

Sortie :

```
{
    "clusters": [
        {
            "status": "ACTIVE",
            "clusterName": "default",
            "registeredContainerInstancesCount": 0,
```

```
            "pendingTasksCount": 0,
            "runningTasksCount": 0,
            "activeServicesCount": 1,
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez Amazon ECS Clusters dans le manuel du développeur Amazon ECS.

Exemple 2 : pour décrire un cluster avec l'option d'attachement

L'`describe-clusters`exemple suivant spécifie l'option ATTACHMENTS. Il récupère les détails du cluster spécifié et une liste des ressources associées au cluster sous forme de pièces jointes. Lorsque vous utilisez un fournisseur de capacité avec un cluster, les ressources, qu'il s'agisse de AutoScaling plans ou de politiques de dimensionnement, seront représentées sous forme de pièces jointes asp ou as_policy.

```
aws ecs describe-clusters \
    --include ATTACHMENTS \
    --clusters sampleCluster
```

Sortie :

```
{
    "clusters": [
        {
            "clusterArn": "arn:aws:ecs:af-south-1:123456789222:cluster/
sampleCluster",
            "clusterName": "sampleCluster",
            "status": "ACTIVE",
            "registeredContainerInstancesCount": 0,
            "runningTasksCount": 0,
            "pendingTasksCount": 0,
            "activeServicesCount": 0,
            "statistics": [],
            "tags": [],
            "settings": [],
            "capacityProviders": [
                "sampleCapacityProvider"
```

```
            ],
            "defaultCapacityProviderStrategy": [],
            "attachments": [
                {
                    "id": "a1b2c3d4-5678-901b-cdef-EXAMPLE22222",
                    "type": "as_policy",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "capacityProviderName",
                            "value": "sampleCapacityProvider"
                        },
                        {
                            "name": "scalingPolicyName",
                            "value": "ECSManagedAutoScalingPolicy-3048e262-
fe39-4eaf-826d-6f975d303188"
                        }
                    ]
                }
            ],
            "attachmentsStatus": "UPDATE_COMPLETE"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez Amazon ECS Clusters dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeClustersà la section Référence des AWS CLI commandes.

`describe-container-instances`

L'exemple de code suivant montre comment utiliser`describe-container-instances`.

AWS CLI

Pour décrire une instance de conteneur

L'`describe-container-instances`exemple suivant récupère les détails d'une instance de conteneur dans le `update` cluster, en utilisant l'UUID de l'instance de conteneur comme identifiant.

```
aws ecs describe-container-instances \
    --cluster update \
    --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :


```
{
    "failures": [],
    "containerInstances": [
        {
            "status": "ACTIVE",
            "registeredResources": [
                {
                    "integerValue": 2048,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "CPU",
                    "doubleValue": 0.0
                },
                {
                    "integerValue": 3955,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "MEMORY",
                    "doubleValue": 0.0
                },
                {
                    "name": "PORTS",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET",
                    "integerValue": 0
                }
            ],
            "ec2InstanceId": "i-A1B2C3D4",
            "agentConnected": true,
```

```
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "pendingTasksCount": 0,
            "remainingResources": [
                {
                    "integerValue": 2048,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "CPU",
                    "doubleValue": 0.0
                },
                {
                    "integerValue": 3955,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "MEMORY",
                    "doubleValue": 0.0
                },
                {
                    "name": "PORTS",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET",
                    "integerValue": 0
                }
            ],
            "runningTasksCount": 0,
            "versionInfo": {
                "agentVersion": "1.0.0",
                "agentHash": "4023248",
                "dockerVersion": "DockerVersion: 1.5.0"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Amazon ECS Container Instances dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeContainerInstancesà la section Référence des AWS CLI commandes.

`describe-service-deployments`

L'exemple de code suivant montre comment utiliser`describe-service-deployments`.

AWS CLI

Pour décrire les détails du déploiement du service

L'`describe-service-deployments`exemple suivant renvoie les détails du déploiement du service avec l'ARN`arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe`.

```
aws ecs describe-service-deployments \
    --service-deployment-arn arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe
```

Sortie :

```
{
    "serviceDeployments": [
        {
            "serviceDeploymentArn": "arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-cluster",
            "createdAt": "2024-10-31T08:03:30.917000-04:00",
            "startedAt": "2024-10-31T08:03:32.510000-04:00",
            "finishedAt": "2024-10-31T08:05:04.527000-04:00",
            "updatedAt": "2024-10-31T08:05:04.527000-04:00",
            "sourceServiceRevisions": [],
            "targetServiceRevision": {
                "arn": "arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678",
                "requestedTaskCount": 1,
```

```
                "runningTaskCount": 1,
                "pendingTaskCount": 0
            },
            "status": "SUCCESSFUL",
            "deploymentConfiguration": {
                "deploymentCircuitBreaker": {
                    "enable": true,
                    "rollback": true
                },
                "maximumPercent": 200,
                "minimumHealthyPercent": 100,
                "alarms": {
                    "alarmNames": [],
                    "rollback": false,
                    "enable": false
                }
            },
            "deploymentCircuitBreaker": {
                "status": "MONITORING_COMPLETE",
                "failureCount": 0,
                "threshold": 3
            },
            "alarms": {
                "status": "DISABLED"
            }
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez Afficher l'historique des services à l'aide des déploiements de services Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeServiceDeploymentsà la section Référence des AWS CLI commandes.

`describe-service-revisions`

L'exemple de code suivant montre comment utiliser`describe-service-revisions`.

AWS CLI

Pour décrire les détails de la révision du service

L'`describe-service-revisions`exemple suivant renvoie les détails de la révision du service pour la révision du service avec l'ARN`arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678`.

```
aws ecs describe-service-revisions \
    --service-revision-arns arn:aws:ecs:us-east-1:123456789012:service-revision/
example-cluster/example-service/1485800978477494678
```

Sortie :

```
{
    "serviceRevisions": [
        {
            "serviceRevisionArn": "arn:aws:ecs:us-east-1:123456789012:service-
revision/example-cluster/example-service/1485800978477494678",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-
cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-
cluster",
            "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
webserver:5",
            "capacityProviderStrategy": [
                {
                    "capacityProvider": "FARGATE",
                    "weight": 1,
                    "base": 0
                }
            ],
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-0d0eab1bb38d5ca64",
                        "subnet-0db5010045995c2d5"
                    ],
                    "securityGroups": [
                        "sg-02556bf85a191f59a"
                    ],
                    "assignPublicIp": "ENABLED"
                }
            },
            "containerImages": [
```

```
                {
                    "containerName": "aws-otel-collector",
                    "imageDigest":
 "sha256:7a1b3560655071bcacd66902c20ebe9a69470d5691fe3bd36baace7c2f3c4640",
                    "image": "public.ecr.aws/aws-observability/aws-otel-
collector:v0.32.0"
                },
                {
                    "containerName": "web",
                    "imageDigest":
 "sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb",
                    "image": "nginx"
                }
            ],
            "guardDutyEnabled": false,
            "serviceConnectConfiguration": {
                "enabled": false
            },
            "createdAt": "2024-10-31T08:03:29.302000-04:00"
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez les révisions du service Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeServiceRevisionsà la section Référence des AWS CLI commandes.

`describe-services`

L'exemple de code suivant montre comment utiliser`describe-services`.

AWS CLI

Pour décrire un service

L'`describe-services`exemple suivant permet de récupérer les détails du `my-http-service` service dans le cluster par défaut.

```
aws ecs describe-services --services my-http-service
```

Sortie :

```
{
    "services": [
        {
            "status": "ACTIVE",
            "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:1",
            "pendingCount": 0,
            "loadBalancers": [],
            "desiredCount": 10,
            "createdAt": 1466801808.595,
            "serviceName": "my-http-service",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/my-http-
service",
            "deployments": [
                {
                    "status": "PRIMARY",
                    "pendingCount": 0,
                    "createdAt": 1466801808.595,
                    "desiredCount": 10,
                    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:1",
                    "updatedAt": 1428326312.703,
                    "id": "ecs-svc/1234567890123456789",
                    "runningCount": 10
                }
            ],
            "events": [
                {
                    "message": "(service my-http-service) has reached a steady
 state.",
                    "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
                    "createdAt": 1466801812.435
                }
            ],
            "runningCount": 10
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Services du guide du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeServicesà la section Référence des AWS CLI commandes.

`describe-task-definition`

L'exemple de code suivant montre comment utiliser`describe-task-definition`.

AWS CLI

Pour décrire une définition de tâche

L'`describe-task-definition`exemple suivant récupère les détails d'une définition de tâche.

```
aws ecs describe-task-definition \
    --task-definition hello_world:8
```

Sortie :

```
{
    "taskDefinition": {
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:8",
        "containerDefinitions": [
            {
                "cpu": 10,
                "environment": [],
                "essential": true,
                "image": "wordpress",
                "links": [
                    "mysql"
                ] ,
                "memory": 500,
                "mountPoints": [],
                "name": "wordpress",
                "portMappings": [
                    {
                        "containerPort": 80,
                        "hostPort": 80
                    }
                ],
```

```
                "volumesFrom": []
            },
            {
                "cpu": 10,
                "environment": [
                    {
                        "name": "MYSQL_ROOT_PASSWORD",
                        "value": "password"
                    }
                ],
                "essential": true,
                "image": "mysql",
                "memory": 500,
                "mountPoints": [],
                "name": "mysql",
                "portMappings": [],
                "volumesFrom": []
            }
        ],
    "family": "hello_world",
    "revision": 8,
    "volumes": [],
    "status": "ACTIVE",
    "placementConstraints": [],
    "compatibilities": [
        "EXTERNAL",
        "EC2"
    ],
    "registeredAt": "2024-06-21T11:15:12.669000-05:00",
    "registeredBy": "arn:aws:sts::012345678910:assumed-role/demo-role/jane-doe"
    },
    "tags": []
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeTaskDefinitionà la section Référence des AWS CLI commandes.

`describe-task-sets`

L'exemple de code suivant montre comment utiliser`describe-task-sets`.

AWS CLI

Pour décrire un ensemble de tâches

L'`describe-task-sets`exemple suivant décrit un ensemble de tâches dans un service qui utilise un déployeur externe.

```
aws ecs describe-task-sets \
    --cluster MyCluster \
    --service MyService \
    --task-sets arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-svc/1234567890123456789
```

Sortie :

```
{
    "taskSets": [
        {
            "id": "ecs-svc/1234567890123456789",
            "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
            "status": "ACTIVE",
            "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
            "computedDesiredCount": 0,
            "pendingCount": 0,
            "runningCount": 0,
            "createdAt": 1557207715.195,
            "updatedAt": 1557207740.014,
            "launchType": "EC2",
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-12344321"
                    ],
                    "securityGroups": [
                        "sg-1234431"
                    ],
                    "assignPublicIp": "DISABLED"
                }
            },
            "loadBalancers": [],
            "serviceRegistries": [],
```

```
            "scale": {
                "value": 0.0,
                "unit": "PERCENT"
            },
            "stabilityStatus": "STEADY_STATE",
            "stabilityStatusAt": 1557207740.014
        }
    ],
    "failures": []
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTaskSetsà la section Référence des AWS CLI commandes.

`describe-tasks`

L'exemple de code suivant montre comment utiliser`describe-tasks`.

AWS CLI

Exemple 1 : Pour décrire les tâches d'une seule tâche

L'`describe-tasks`exemple suivant permet de récupérer les détails d'une tâche dans un cluster. Vous pouvez spécifier la tâche en utilisant son ID ou son ARN complet. Cet exemple utilise l'ARN complet de la tâche.

```
aws ecs describe-tasks \
    --cluster MyCluster \
    --tasks arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE
```

Sortie :

```
{
    "tasks": [
        {
            "attachments": [],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
```

```
            ],
            "availabilityZone": "us-east-1b",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
            "connectivity": "CONNECTED",
            "connectivityAt": "2021-08-11T12:21:26.681000-04:00",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/test/025c7e2c5e054a6790a29fc1fEXAMPLE",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/4d590253bb114126b7afa7b58eea9221/a992d1cc-ea46-474a-b6e8-24688EXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE",
                    "name": "simple-app",
                    "image": "httpd:2.4",
                    "runtimeId":
 "91251eed27db90006ad67b1a08187290869f216557717dd5c39b37c94EXAMPLE",
                    "lastStatus": "RUNNING",
                    "networkBindings": [
                        {
                            "bindIP": "0.0.0.0",
                            "containerPort": 80,
                            "hostPort": 80,
                            "protocol": "tcp"
                        }
                    ],
                    "networkInterfaces": [],
                    "healthStatus": "UNKNOWN",
                    "cpu": "10",
                    "memory": "300"
                }
            ],
            "cpu": "10",
            "createdAt": "2021-08-11T12:21:26.681000-04:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "service:testupdate",
            "healthStatus": "UNKNOWN",
            "lastStatus": "RUNNING",
            "launchType": "EC2",
            "memory": "300",
            "overrides": {
                "containerOverrides": [
                    {
```

```
                        "name": "simple-app"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "pullStartedAt": "2021-08-11T12:21:28.234000-04:00",
            "pullStoppedAt": "2021-08-11T12:21:33.793000-04:00",
            "startedAt": "2021-08-11T12:21:34.945000-04:00",
            "startedBy": "ecs-svc/968695068243EXAMPLE",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58eea9221",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/console-sample-app-static2:1",
            "version": 2
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

Exemple 2 : Pour décrire plusieurs tâches

L'`describe-tasks`exemple suivant permet de récupérer les détails de plusieurs tâches d'un cluster. Vous pouvez spécifier la tâche en utilisant son ID ou son ARN complet. Cet exemple utilise l'ensemble IDs des tâches.

```
aws ecs describe-tasks \
    --cluster MyCluster \
    --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"
```

Sortie :

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
                    "type": "ElasticNetworkInterface",
                    "status": "ATTACHED",
```

```
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-0d0eab1bb3EXAMPLE"
                        },
                        {
                            "name": "networkInterfaceId",
                            "value": "eni-0fa40520aeEXAMPLE"
                        },
                        {
                            "name": "macAddress",
                            "value": "0e:89:76:28:07:b3"
                        },
                        {
                            "name": "privateDnsName",
                            "value": "ip-10-0-1-184.ec2.internal"
                        },
                        {
                            "name": "privateIPv4Address",
                            "value": "10.0.1.184"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
            "connectivity": "CONNECTED",
            "connectivityAt": "2021-12-20T12:13:37.875000-05:00",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE/aad3ba00-83b3-4dac-84d4-11f8cEXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
                    "name": "web",
                    "image": "nginx",
                    "runtimeId": "74de0355a10a4f979ac495c14EXAMPLE-265927825",
                    "lastStatus": "RUNNING",
```

```
                    "networkBindings": [],
                    "networkInterfaces": [
                        {
                            "attachmentId": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
                            "privateIpv4Address": "10.0.1.184"
                        }
                    ],
                    "healthStatus": "UNKNOWN",
                    "cpu": "99",
                    "memory": "100"
                }
            ],
            "cpu": "256",
            "createdAt": "2021-12-20T12:13:20.226000-05:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "service:tdsevicetag",
            "healthStatus": "UNKNOWN",
            "lastStatus": "RUNNING",
            "launchType": "FARGATE",
            "memory": "512",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "web"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "pullStartedAt": "2021-12-20T12:13:42.665000-05:00",
            "pullStoppedAt": "2021-12-20T12:13:46.543000-05:00",
            "startedAt": "2021-12-20T12:13:48.086000-05:00",
            "startedBy": "ecs-svc/988401040018EXAMPLE",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
            "version": 3,
            "ephemeralStorage": {
            "sizeInGiB": 20
            }
```

```
        },
        {
            "attachments": [
                {
                    "id": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
                    "type": "ElasticNetworkInterface",
                    "status": "ATTACHED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-0d0eab1bb3EXAMPLE"
                        },
                        {
                            "name": "networkInterfaceId",
                            "value": "eni-064c7766daEXAMPLE"
                        },
                        {
                            "name": "macAddress",
                            "value": "0e:76:83:01:17:a9"
                        },
                        {
                            "name": "privateDnsName",
                            "value": "ip-10-0-1-41.ec2.internal"
                        },
                        {
                            "name": "privateIPv4Address",
                            "value": "10.0.1.41"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
            "connectivity": "CONNECTED",
            "connectivityAt": "2021-12-20T12:13:35.243000-05:00",
            "containers": [
                {
```

```
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/d789e94343414c25b9f6bd59eEXAMPLE/9afef792-609b-43a5-bb6a-3efdbEXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
                    "name": "web",
                    "image": "nginx",
                    "runtimeId": "d789e94343414c25b9f6bd59eEXAMPLE-265927825",
                    "lastStatus": "RUNNING",
                    "networkBindings": [],
                    "networkInterfaces": [
                        {
                            "attachmentId": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
                            "privateIpv4Address": "10.0.1.41"
                        }
                    ],
                    "healthStatus": "UNKNOWN",
                    "cpu": "99",
                    "memory": "100"
                }
            ],
            "cpu": "256",
            "createdAt": "2021-12-20T12:13:20.226000-05:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "service:tdsevicetag",
            "healthStatus": "UNKNOWN",
            "lastStatus": "RUNNING",
            "launchType": "FARGATE",
            "memory": "512",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "web"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "pullStartedAt": "2021-12-20T12:13:44.611000-05:00",
            "pullStoppedAt": "2021-12-20T12:13:48.251000-05:00",
            "startedAt": "2021-12-20T12:13:49.326000-05:00",
            "startedBy": "ecs-svc/988401040018EXAMPLE",
            "tags": [],
```

```
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
            "version": 3,
            "ephemeralStorage": {
                "sizeInGiB": 20
            }
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous DescribeTasksà la section Référence des AWS CLI commandes.

`execute-command`

L'exemple de code suivant montre comment utiliser`execute-command`.

AWS CLI

Pour exécuter une commande interactive /bin/sh

L'`execute-command`exemple suivant exécute une commande interactive /bin/sh sur un conteneur nommé MyContainer d'après une tâche dont l'identifiant est. `arn:aws:ecs:us-east-1:123456789012:task/MyCluster/d789e94343414c25b9f6bd59eEXAMPLE`

```
aws ecs execute-command \
    --cluster MyCluster \
    --task arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE \
    --container MyContainer \
    --interactive \
    --command "/bin/sh"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation d'Amazon ECS Exec pour le débogage dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ExecuteCommandà la section Référence des AWS CLI commandes.

`get-task-protection`

L'exemple de code suivant montre comment utiliser`get-task-protection`.

AWS CLI

Récupérer l'état de protection de la tâche dans le service ECS

Vous trouverez `get-task-protection` ci-dessous l'état de protection des tâches ECS appartenant au service Amazon ECS.

```
aws ecs get-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24
```

Sortie :

```
{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

Pour en savoir plus sur la protection des tâches, consultez la section Protéger vos tâches Amazon ECS contre l'interruption à la suite d'événements évolutifs dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetTaskProtectionà la section Référence des AWS CLI commandes.

`list-account-settings`

L'exemple de code suivant montre comment utiliser`list-account-settings`.

AWS CLI

Exemple 1 : pour consulter les paramètres d'un compte

L'`list-account-settings`exemple suivant montre les paramètres effectifs d'un compte.

```
aws ecs list-account-settings --effective-settings
```

Sortie :

```
{
    "settings": [
        {
            "name": "containerInstanceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:root"
        },
        {
            "name": "serviceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:root"
        },
        {
            "name": "taskLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:root"
        }
    ]
}
```

Exemple 2 : pour afficher les paramètres du compte pour un utilisateur ou un rôle IAM spécifique

L'`list-account-settings`exemple suivant affiche les paramètres du compte pour l'utilisateur IAM ou le rôle IAM spécifié.

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Sortie :

```
{
    "settings": [
        {
            "name": "serviceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:user/MyUser"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Resource Names (ARNs) et IDs le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListAccountSettingsà la section Référence des AWS CLI commandes.

`list-attributes`

L'exemple de code suivant montre comment utiliser`list-attributes`.

AWS CLI

Pour répertorier les instances de conteneur contenant un attribut spécifique

L'exemple suivant répertorie les attributs des instances de conteneur dont l'`stack=production`attribut se trouve dans le cluster par défaut.

```
aws ecs list-attributes \
    --target-type container-instance \
    --attribute-name stack \
    --attribute-value production \
    --cluster default
```

Sortie :

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
```

```
            "value": "production"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration de l'agent de conteneur Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListAttributesà la section Référence des AWS CLI commandes.

`list-clusters`

L'exemple de code suivant montre comment utiliser`list-clusters`.

AWS CLI

Pour répertorier vos clusters disponibles

L'`list-clusters`exemple suivant répertorie tous les clusters disponibles.

```
aws ecs list-clusters
```

Sortie :

```
{
    "clusterArns": [
        "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
        "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
    ]
}
```

Pour plus d'informations, consultez Amazon ECS Clusters dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListClustersà la section Référence des AWS CLI commandes.

`list-container-instances`

L'exemple de code suivant montre comment utiliser`list-container-instances`.

AWS CLI

Pour répertorier les instances de conteneur dans un cluster

L'`list-container-instances`exemple suivant répertorie toutes les instances de conteneur disponibles dans un cluster.

```
aws ecs list-container-instances --cluster MyCluster
```

Sortie :

```
{
    "containerInstanceArns": [
        "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
    ]
}
```

Pour plus d'informations, consultez la section Amazon ECS Container Instances dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListContainerInstancesà la section Référence des AWS CLI commandes.

`list-service-deployments`

L'exemple de code suivant montre comment utiliser`list-service-deployments`.

AWS CLI

Pour répertorier les déploiements de services

L'`list-service-deployments`exemple suivant récupère les déploiements de service pour le service nommé. `example-service`

```
aws ecs list-service-deployments \
    --service arn:aws:ecs:us-east-1:123456789012:service/example-cluster/example-
service
```

Sortie :

```
{
    "serviceDeployments": [
        {
            "serviceDeploymentArn": "arn:aws:ecs:us-east-1:123456789012:service-
deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-
cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-
cluster",
            "startedAt": "2024-10-31T08:03:32.510000-04:00",
            "createdAt": "2024-10-31T08:03:30.917000-04:00",
            "finishedAt": "2024-10-31T08:05:04.527000-04:00",
            "targetServiceRevisionArn": "arn:aws:ecs:us-east-1:123456789012:service-
revision/example-cluster/example-service/1485800978477494678",
            "status": "SUCCESSFUL"
        }
    ]
}
```

Pour plus d'informations, consultez Afficher l'historique des services à l'aide des déploiements de services Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListServiceDeploymentsà la section Référence des AWS CLI commandes.

`list-services-by-namespace`

L'exemple de code suivant montre comment utiliser`list-services-by-namespace`.

AWS CLI

Pour répertorier les services dans un espace de noms

L'`list-services-by-namespace`exemple suivant répertorie tous les services configurés pour l'espace de noms spécifié dans votre région par défaut.

```
aws ecs list-services-by-namespace \
    --namespace service-connect
```

Sortie :

```
{
```

```
    "serviceArns": [
        "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",
        "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-
service"
    ]
}
```

Pour plus d'informations, consultez Service Connect dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListServicesByNamespaceà la section Référence des AWS CLI commandes.

`list-services`

L'exemple de code suivant montre comment utiliser`list-services`.

AWS CLI

Pour répertorier les services d'un cluster

L'`list-services`exemple suivant montre comment répertorier les services exécutés dans un cluster.

```
aws ecs list-services --cluster MyCluster
```

Sortie :

```
 {
     "serviceArns": [
         "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"
     ]
}
```

Pour plus d'informations, consultez la section Services du guide du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListServicesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises d'un cluster spécifique.

```
aws ecs list-tags-for-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

Sortie :

```
{
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        },
        {
            "key": "key3",
            "value": "value3"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-task-definition-families`

L'exemple de code suivant montre comment utiliser`list-task-definition-families`.

AWS CLI

Exemple 1 : pour répertorier les familles de définitions de tâches enregistrées

L'`list-task-definition-families`exemple suivant répertorie toutes les familles de définitions de tâches enregistrées.

```
aws ecs list-task-definition-families
```

Sortie :

```
{
    "families": [
        "node-js-app",
        "web-timer",
        "hpcc",
        "hpcc-c4-8xlarge"
    ]
}
```

Exemple 2 : pour filtrer les familles de définitions de tâches enregistrées

L'`list-task-definition-families`exemple suivant répertorie les révisions des définitions de tâches qui commencent par « hpcc ».

```
aws ecs list-task-definition-families --family-prefix hpcc
```

Sortie :

```
{
    "families": [
        "hpcc",
        "hpcc-c4-8xlarge"
    ]
}
```

Pour plus d'informations, consultez la section Paramètres de définition des tâches dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTaskDefinitionFamiliesà la section Référence des AWS CLI commandes.

`list-task-definitions`

L'exemple de code suivant montre comment utiliser`list-task-definitions`.

AWS CLI

Exemple 1 : pour répertorier les définitions de tâches enregistrées

L'`list-task-definitions`exemple suivant répertorie toutes les définitions de tâches enregistrées.

```
aws ecs list-task-definitions
```

Sortie :

```
{
    "taskDefinitionArns": [
        "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep300:2",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep360:1",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
    ]
}
```

Exemple 2 : pour répertorier les définitions de tâches enregistrées dans une famille

L' list-task-definitionsexemple suivant répertorie les révisions des définitions de tâches d'une famille spécifiée.

```
aws ecs list-task-definitions --family-prefix wordpress
```

Sortie :

```
{
    "taskDefinitionArns": [
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
    ]
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListTaskDefinitionsà la section Référence des AWS CLI commandes.

`list-tasks`

L'exemple de code suivant montre comment utiliser`list-tasks`.

AWS CLI

Exemple 1 : pour répertorier les tâches d'un cluster

L'`list-tasks`exemple suivant répertorie toutes les tâches d'un cluster.

```
aws ecs list-tasks --cluster default
```

Sortie :

```
{
    "taskArns": [
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-11111EXAMPLE",
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE"
    ]
}
```

Exemple 2 : pour répertorier les tâches sur une instance de conteneur particulière

L'`list-tasks`exemple suivant répertorie les tâches sur une instance de conteneur, en utilisant l'UUID de l'instance de conteneur comme filtre.

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

Sortie :

```
{
    "taskArns": [
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE"
```

```
    ]
}
```

Pour plus d'informations, consultez les définitions des tâches Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous ListTasksà la section Référence des AWS CLI commandes.

`put-account-setting-default`

L'exemple de code suivant montre comment utiliser`put-account-setting-default`.

AWS CLI

Pour modifier les paramètres du compte par défaut

L'`put-account-setting-default`exemple suivant modifie le paramètre de compte par défaut pour tous les utilisateurs ou rôles IAM de votre compte. Ces modifications s'appliquent à l'ensemble du AWS compte, sauf si un utilisateur ou un rôle IAM remplace explicitement ces paramètres pour lui-même.

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

Sortie :

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:root"
    }
}
```

Pour plus d'informations, consultez Amazon Resource Names (ARNs) et IDs le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous PutAccountSettingDefaultà la section Référence des AWS CLI commandes.

`put-account-setting`

L'exemple de code suivant montre comment utiliser`put-account-setting`.

AWS CLI

Pour modifier les paramètres de votre compte utilisateur IAM

L'`put-account-setting`exemple suivant définit le paramètre du `containerInsights` compte sur `enhanced` votre compte utilisateur IAM. Cela active Container Insights avec une observabilité améliorée.

```
aws ecs put-account-setting \
    --name containerInsights \
    --value enhanced
```

Sortie :

```
{
    "setting": {
        "name": "containerInsights",
        "value": "enhanced",
        "principalArn": "arn:aws:iam::123456789012:user/johndoe",
        "type": "user"
    }
}
```

Pour plus d'informations, consultez la section Modification des paramètres du compte dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutAccountSettingà la section Référence des AWS CLI commandes.

`put-account-settings`

L'exemple de code suivant montre comment utiliser`put-account-settings`.

AWS CLI

Pour modifier les paramètres du compte pour un utilisateur ou un rôle IAM

L'`put-account-setting`exemple suivant modifie les paramètres du compte pour l'utilisateur IAM ou le rôle IAM spécifié.

```
aws ecs put-account-setting \
    --name serviceLongArnFormat \
    --value enabled \
    --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Sortie :

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutAccountSettingsà la section Référence des AWS CLI commandes.

`put-attributes`

L'exemple de code suivant montre comment utiliser`put-attributes`.

AWS CLI

Pour créer un attribut et l'associer à une ressource Amazon ECS

Ce qui suit `put-attributes` applique un attribut avec la pile de noms et la production de valeur à une instance de conteneur.

```
aws ecs put-attributes \
    --attributes name=stack,value=production,targetId=arn:aws:ecs:us-
west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Sortie :

```
{
    "attributes": [
        {
```

```
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutAttributesà la section Référence des AWS CLI commandes.

`put-cluster-capacity-providers`

L'exemple de code suivant montre comment utiliser`put-cluster-capacity-providers`.

AWS CLI

Exemple 1 : pour ajouter un fournisseur de capacité existant à un cluster

L'`put-cluster-capacity-providers`exemple suivant ajoute un fournisseur de capacité existant à un cluster. La `create-capacity-provider` commande est utilisée pour créer un fournisseur de capacité. La `describe-clusters` commande est utilisée pour décrire les fournisseurs de capacité actuels et la stratégie de fournisseur de capacité par défaut associée à un cluster. Lorsque vous ajoutez un nouveau fournisseur de capacité à un cluster, vous devez spécifier tous les fournisseurs de capacité existants en plus du nouveau fournisseur de capacité que vous souhaitez associer au cluster. Vous devez également spécifier la stratégie de fournisseur de capacité par défaut à associer au cluster. Dans cet exemple, le fournisseur de `MyCapacityProvider1` capacité est associé au `MyCluster` cluster et vous souhaitez ajouter le fournisseur de `MyCapacityProvider2` capacité et l'inclure dans la stratégie de fournisseur de capacité par défaut afin que les tâches soient réparties uniformément entre les deux fournisseurs de capacité.

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1",
            "MyCapacityProvider2"
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "MyCapacityProvider1",
                "weight": 1,
                "base": 0
            },
            {
                "capacityProvider": "MyCapacityProvider2",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
```

```
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour supprimer un fournisseur de capacité d'un cluster

L'`put-cluster-capacity-providers`exemple suivant supprime un fournisseur de capacité d'un cluster. La `describe-clusters` commande est utilisée pour décrire les fournisseurs de capacité actuels associés à un cluster. Lorsque vous supprimez un fournisseur de capacité d'un cluster, vous devez spécifier les fournisseurs de capacité que vous souhaitez voir associés au cluster ainsi que la stratégie de fournisseur de capacité par défaut à associer au cluster. Dans cet exemple, les fournisseurs de `MyCapacityProvider2` capacité `MyCapacityProvider1` et sont associés au cluster et vous souhaitez supprimer le fournisseur de `MyCapacityProvider2` capacité. Vous ne le spécifiez donc que `MyCapacityProvider1` dans la commande avec la stratégie de fournisseur de capacité par défaut mise à jour.

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers MyCapacityProvider1 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1"
        ],
        "defaultCapacityProviderStrategy": [
            "capacityProvider": "MyCapacityProvider1",
            "weight": 1,
            "base": 0
        ],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
```

```
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "DELETING",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

Exemple 3 : pour supprimer tous les fournisseurs de capacité d'un cluster

L'`put-cluster-capacity-providers`exemple suivant supprime tous les fournisseurs de capacité existants du cluster.

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers [] \
    --default-capacity-provider-strategy []
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [],
        "defaultCapacityProviderStrategy": [],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "DELETING",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "DELETING",
                "details": [
                    {
                        "name": "capacityProviderName",
```

```
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

Pour plus d'informations, consultez la section Fournisseurs de capacité de cluster dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutClusterCapacityProvidersà la section Référence des AWS CLI commandes.

`register-task-definition`

L'exemple de code suivant montre comment utiliser`register-task-definition`.

AWS CLI

Exemple 1 : pour enregistrer une définition de tâche dans un fichier JSON

L'`register-task-definition`exemple suivant enregistre une définition de tâche dans la famille spécifiée. Les définitions de conteneur sont enregistrées au format JSON à l'emplacement de fichier spécifié.

```
aws ecs register-task-definition \
    --cli-input-json file://<path_to_json_file>/sleep360.json
```

Contenu de `sleep360.json` :

```
{
    "containerDefinitions": [
        {
            "name": "sleep",
            "image": "busybox",
```

```
            "cpu": 10,
            "command": [
                "sleep",
                "360"
            ],
            "memory": 10,
            "essential": true
        }
    ],
    "family": "sleep360"
}
```

Sortie :

```
{
    "taskDefinition": {
        "status": "ACTIVE",
        "family": "sleep360",
        "placementConstraints": [],
        "compatibilities": [
                "EXTERNAL",
                "EC2"
        ],
        "volumes": [],
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
sleep360:1",
        "containerDefinitions": [
            {
                "environment": [],
                "name": "sleep",
                "mountPoints": [],
                "image": "busybox",
                "cpu": 10,
                "portMappings": [],
                "command": [
                    "sleep",
                    "360"
                ],
        "memory": 10,
        "essential": true,
        "volumesFrom": []
        }
    ],
```

```
        "revision": 1
    }
}
```

Pour plus d'informations, consultez des exemples de définitions de tâches dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour enregistrer une définition de tâche avec un paramètre de chaîne JSON

L'`register-task-definition`exemple suivant enregistre une définition de tâche à l'aide de définitions de conteneur fournies sous forme de paramètre de chaîne JSON avec des guillemets évités.

```
aws ecs register-task-definition \
    --family sleep360 \
    --container-definitions "[{\"name\":\"sleep\",\"image\":\"busybox\",\"cpu\":10,
\"command\":[\"sleep\",\"360\"],\"memory\":10,\"essential\":true}]"
```

Le résultat est identique à celui de l'exemple précédent.

Pour plus d'informations, consultez la section Création d'une définition de tâche dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RegisterTaskDefinitionà la section Référence des AWS CLI commandes.

`run-task`

L'exemple de code suivant montre comment utiliser`run-task`.

AWS CLI

Exemple 1 : pour exécuter une tâche sur votre cluster par défaut

L'`run-task`exemple suivant exécute une tâche sur le cluster par défaut et utilise un jeton client.

```
aws ecs run-task \
    --cluster default \
    --task-definition sleep360:1 \
    --client-token 550e8400-e29b-41d4-a716-446655440000
```

Sortie :


```
{
    "tasks": [
        {
            "attachments": [],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
            "capacityProviderName": "example-capacity-provider",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/default",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/default/bc4d2ec611d04bb7bb97e83ceEXAMPLE",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
default/d6f51cc5bbc94a47969c92035e9f66f8/75853d2d-711e-458a-8362-0f0aEXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
                    "name": "sleep",
                    "image": "busybox",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "10",
                    "memory": "10"
                }
            ],
            "cpu": "10",
            "createdAt": "2023-11-21T16:59:34.403000-05:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:sleep360",
            "lastStatus": "PENDING",
            "launchType": "EC2",
            "memory": "10",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
```

```
                ],
                "inferenceAcceleratorOverrides": []
            },
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/sleep360:1",
            "version": 1
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Exécution d'une application en tant que tâche autonome dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour configurer un volume Amazon EBS pour une tâche autonome

L'`run-task`exemple suivant configure un volume Amazon EBS chiffré pour une tâche Fargate sur le cluster par défaut. Vous devez avoir un rôle d'infrastructure Amazon ECS configuré avec la politique `AmazonECSInfrastructureRolePolicyForVolumes` gérée attachée. Vous devez spécifier une définition de tâche portant le même nom de volume que dans la `run-task` demande. Cet exemple utilise l'`--cli-input-json`option et un fichier d'entrée JSON appelé`ebs.json`.

```
aws ecs run-task \
    --cli-input-json file://ebs.json
```

Contenu de `ebs.json` :

```
{
   "cluster": "default",
   "taskDefinition": "mytaskdef",
   "launchType": "FARGATE",
   "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
```

```
   "volumeConfigurations": [
        {
            "name": "myEBSVolume",
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam::1111222333:role/ecsInfrastructureRole",
                "encrypted": true,
                "kmsKeyId":
 "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        }
    ]
}
```

Sortie :

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "ce868693-15ca-4083-91ac-f782f64000c9",
                    "type": "ElasticNetworkInterface",
                    "status": "PRECREATED",
                    "details": [
                        {
                        "name": "subnetId",
                        "value": "subnet-070982705451dad82"
                        }
                    ]
                },
                {
                    "id": "a17ed863-786c-4372-b5b3-b23e53f37877",
                    "type": "AmazonElasticBlockStorage",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "roleArn",
                            "value": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole"
                        },
                        {
```

```
                            "name": "volumeName",
                            "value": "myEBSVolume"
                        },
                        {
                            "name": "deleteOnTermination",
                            "value": "true"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-west-2b",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-west-2:123456789012:container/
default/7f1fbd3629434cc4b82d72d2f09b67c9/e21962a2-f328-4699-98a3-5161ac2c186a",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/7f1fbd3629434cc4b82d72d2f09b67c9",
                    "name": "container-using-ebs",
                    "image": "amazonlinux:2",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "0"
                }
            ],
            "cpu": "1024",
            "createdAt": "2025-01-23T10:29:46.650000-06:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:mytaskdef",
            "lastStatus": "PROVISIONING",
            "launchType": "FARGATE",
            "memory": "3072",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "container-using-ebs"
                    }
```

```
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/7f1fbd3629434cc4b82d72d2f09b67c9",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:4",
            "version": 1,
            "ephemeralStorage": {
                "sizeInGiB": 20
            },
            "fargateEphemeralStorage": {
                "sizeInGiB": 20
            }
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Utiliser les volumes Amazon EBS avec Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous RunTaskà la section Référence des AWS CLI commandes.

`start-task`

L'exemple de code suivant montre comment utiliser`start-task`.

AWS CLI

Exemple 1 : pour démarrer une nouvelle tâche

L'`start-task`exemple suivant démarre une tâche en utilisant la dernière révision de la définition de `sleep360` tâche sur l'instance de conteneur spécifiée dans le cluster par défaut.

```
aws ecs start-task \
    --task-definition sleep360 \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

Sortie :

```
{
    "tasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:3",
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
                ]
            },
            "lastStatus": "PENDING",
            "desiredStatus": "RUNNING",
            "cpu": "128",
            "memory": "128",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-
west-2:123456789012:container/75f11ed4-8a3d-4f26-a33b-ad1db9e02d41",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
                    "name": "sleep",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "10",
                    "memory": "10"
                }
            ],
            "version": 1,
            "createdAt": 1563421494.186,
            "group": "family:sleep360",
            "launchType": "EC2",
            "attachments": [],
            "tags": []
        }
    ],
```

```
    "failures": []
}
```

Pour plus d'informations, consultez Planifier vos conteneurs sur Amazon ECS dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour configurer un volume Amazon EBS au début de la tâche

L'`start-task`exemple suivant configure un volume Amazon EBS chiffré pour une tâche sur l'instance de conteneur spécifiée. Vous devez avoir un rôle d'infrastructure Amazon ECS configuré avec la politique `AmazonECSInfrastructureRolePolicyForVolumes` gérée attachée. Vous devez spécifier une définition de tâche portant le même nom de volume que dans la `start-task` demande. Cet exemple utilise l'`--cli-input-json`option et un fichier d'entrée JSON appelé `ebs.json` avec le contenu suivant.

```
aws ecs start-task \
    --cli-input-json file://ebs.json \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

Contenu de `ebs.json` :

```
{
   "cluster": "default",
   "taskDefinition": "mytaskdef",
   "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
   "volumeConfigurations": [
        {
            "name": "myEBSVolume",
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "encrypted": true,
                "kmsKeyId":
 "arn:aws:kms:region:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
```

```
        }
    ]
}
```

Sortie :

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "aea29489-9dcd-49f1-8164-4d91566e1113",
                    "type": "ElasticNetworkInterface",
                    "status": "PRECREATED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-12344321"
                        }
                    ]
                },
                {
                    "id": "f29e1222-9a1e-410f-b499-a12a7cd6d42e",
                    "type": "AmazonElasticBlockStorage",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "roleArn",
                            "value": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole"
                        },
                        {
                            "name": "volumeName",
                            "value": "myEBSVolume"
                        },
                        {
                            "name": "deleteOnTermination",
                            "value": "true"
                        }
                    ]
                }
            ],
            "attributes": [
```

```
                {
                    "name": "ecs.cpu-architecture",
                    "value": "arm64"
                }
            ],
            "availabilityZone": "us-west-2c",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-west-2:123456789012:container/
default/bb122ace3ed84add92c00a351a03c69e/a4a9ed10-51c7-4567-9653-50e71b94f867",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/default/
bb122ace3ed84add92c00a351a03c69e",
                    "name": "container-using-ebs",
                    "image": "amazonlinux:2",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "0"
                }
            ],
            "cpu": "1024",
            "createdAt": "2025-01-23T14:51:05.191000-06:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:mytaskdef",
            "lastStatus": "PROVISIONING",
            "launchType": "EC2",
            "memory": "3072",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "container-using-ebs"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
             "tags": [],
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/default/
bb122ace3ed84add92c00a351a03c69e",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:4",
            "version": 1
```

```
        }
    ],
    "failures": []
}
```

Pour plus d'informations, consultez la section Utiliser les volumes Amazon EBS avec Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous StartTaskà la section Référence des AWS CLI commandes.

`stop-task`

L'exemple de code suivant montre comment utiliser`stop-task`.

AWS CLI

Pour arrêter une tâche

Ce qui suit `stop-task` arrête l'exécution de la tâche spécifiée dans le cluster par défaut.

```
aws ecs stop-task \
    --task 666fdccc2e2d4b6894dd422f4eeee8f8
```

Sortie :

```
{
    "task": {
        "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
        "clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",
        "taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-definition/
sleep360:3",
        "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
        "overrides": {
            "containerOverrides": []
        },
        "lastStatus": "STOPPED",
        "desiredStatus": "STOPPED",
        "cpu": "128",
        "memory": "128",
        "containers": [],
```

```
        "version": 2,
        "stoppedReason": "Taskfailedtostart",
        "stopCode": "TaskFailedToStart",
        "connectivity": "CONNECTED",
        "connectivityAt": 1563421494.186,
        "pullStartedAt": 1563421494.252,
        "pullStoppedAt": 1563421496.252,
        "executionStoppedAt": 1563421497,
        "createdAt": 1563421494.186,
        "stoppingAt": 1563421497.252,
        "stoppedAt": 1563421497.252,
        "group": "family:sleep360",
        "launchType": "EC2",
        "attachments": [],
        "tags": []
    }
}
```

- Pour plus de détails sur l'API, reportez-vous StopTaskà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute une seule balise à la ressource spécifiée.

```
aws ecs tag-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
    --tags key=key1,value=value1
```

Cette commande ne produit aucun résultat.

Pour ajouter plusieurs balises à une ressource

L'`tag-resource`exemple suivant ajoute plusieurs balises à la ressource spécifiée.

```
aws ecs tag-resource \
```

```
--resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
--tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime les balises répertoriées de la ressource spécifiée.

```
aws ecs untag-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
    --tag-keys key1,key2
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-cluster-settings`

L'exemple de code suivant montre comment utiliser`update-cluster-settings`.

AWS CLI

Pour modifier les paramètres de votre cluster

L'`update-cluster-settings`exemple suivant active CloudWatch Container Insights avec une meilleure observabilité pour le `MyCluster` cluster.

```
aws ecs update-cluster-settings \
    --cluster MyCluster \
    --settings name=containerInsights,value=enhanced
```

Sortie :

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-esat-1:123456789012:cluster/MyCluster",
        "clusterName": "default",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enhanced"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Modification des paramètres du compte dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateClusterSettingsà la section Référence des AWS CLI commandes.

`update-cluster`

L'exemple de code suivant montre comment utiliser`update-cluster`.

AWS CLI

Exemple 1 : Mettre à jour le cluster ECS en activant ContainerInsights

Ce qui suit `update-cluster` met à jour la valeur ContainerInsights `enabled` dans un cluster déjà créé. Par défaut, elle est désactivée.

```
aws ecs update-cluster \
    --cluster ECS-project-update-cluster \
    --settings name=containerInsights,value=enabled
```

Sortie :

```
"cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/ECS-project-update-
cluster",
    "clusterName": "ECS-project-update-cluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
        {
            "name": "containerInsights",
            "value": "enabled"
        }
    ],
    "capacityProviders": [
        "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
    ],
    "defaultCapacityProviderStrategy": [
        {
            "capacityProvider": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
            "weight": 1,
            "base": 0
        }
    ],
    "attachments": [
        {
            "id": "069d002b-7634-42e4-b1d4-544f4c8f6380",
            "type": "as_policy",
            "status": "CREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
                },
                {
                    "name": "scalingPolicyName",
```

```
                    "value": "ECSManagedAutoScalingPolicy-152363a6-8c65-484c-
b721-42c3e070ae93"
                }
            ]
        },
        {
            "id": "08b5b6ca-45e9-4209-a65d-e962a27c490a",
            "type": "managed_draining",
            "status": "CREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
                },
                {
                    "name": "autoScalingLifecycleHookName",
                    "value": "ecs-managed-draining-termination-hook"
                }
            ]
        },
        {
            "id": "45d0b36f-8cff-46b6-9380-1288744802ab",
            "type": "sc",
            "status": "ATTACHED",
            "details": []
        }
    ],
    "attachmentsStatus": "UPDATE_COMPLETE",
    "serviceConnectDefaults": {
        "namespace": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
igwrsylmy3kwvcdx"
    }
}
```

Exemple 2 : Mettre à jour le cluster ECS pour définir un espace de noms Service Connect par défaut

Ce qui suit `update-cluster` met à jour le cluster ECS en définissant un espace de noms Service Connect par défaut.

```
aws ecs update-cluster \
    --cluster ECS-project-update-cluster \
```

```
    --service-connect-defaults namespace=test
```

Sortie :

```
 {
     "cluster": {
         "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/ECS-project-
update-cluster",
         "clusterName": "ECS-project-update-cluster",
         "status": "ACTIVE",
         "registeredContainerInstancesCount": 0,
         "runningTasksCount": 0,
         "pendingTasksCount": 0,
         "activeServicesCount": 0,
         "statistics": [],
         "tags": [],
         "settings": [
             {
                 "name": "containerInsights",
                 "value": "enabled"
             }
         ],
         "capacityProviders": [
             "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
         ],
         "defaultCapacityProviderStrategy": [
             {
                 "capacityProvider": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
                 "weight": 1,
                 "base": 0
             }
         ],
         "attachments": [
             {
                 "id": "069d002b-7634-42e4-b1d4-544f4c8f6380",
                 "type": "as_policy",
                 "status": "CREATED",
                 "details": [
                     {
                         "name": "capacityProviderName",
```

```
                        "value": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-152363a6-8c65-484c-
b721-42c3e070ae93"
                    }
                ]
            },
            {
                "id": "08b5b6ca-45e9-4209-a65d-e962a27c490a",
                "type": "managed_draining",
                "status": "CREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
                    },
                    {
                        "name": "autoScalingLifecycleHookName",
                        "value": "ecs-managed-draining-termination-hook"
                    }
                ]
            },
            {
                "id": "45d0b36f-8cff-46b6-9380-1288744802ab",
                "type": "sc",
                "status": "DELETED",
                "details": []
            },
            {
                "id": "3e6890c3-609c-4832-91de-d6ca891b3ef1",
                "type": "sc",
                "status": "ATTACHED",
                "details": []
            },
            {
                "id": "961b8ec1-c2f1-4070-8495-e669b7668e90",
                "type": "sc",
                "status": "DELETED",
                "details": []
            }
```

```
        ],
        "attachmentsStatus": "UPDATE_COMPLETE",
        "serviceConnectDefaults": {
            "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-dtjmxqpfi46ht7dr"
        }
    }
}
```

Pour plus d'informations sur Service Connect, consultez la section Utiliser Service Connect pour connecter les services Amazon ECS avec des noms abrégés dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateClusterà la section Référence des AWS CLI commandes.

`update-container-agent`

L'exemple de code suivant montre comment utiliser`update-container-agent`.

AWS CLI

Pour mettre à jour l'agent de conteneur sur une instance de conteneur Amazon ECS

L'`update-container-agent`exemple suivant met à jour l'agent de conteneur sur l'instance de conteneur spécifiée dans le cluster par défaut.

```
aws ecs update-container-agent --cluster default --container-
instance a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
    "containerInstance": {
        "status": "ACTIVE",
...
        "agentUpdateStatus": "PENDING",
        "versionInfo": {
            "agentVersion": "1.0.0",
            "agentHash": "4023248",
            "dockerVersion": "DockerVersion: 1.5.0"
        }
```

```
    }
}
```

Pour plus d'informations, consultez la section Mise à jour de l'agent de conteneur Amazon ECS dans le manuel du développeur Amazon ECS.

- Pour plus de détails sur l'API, reportez-vous UpdateContainerAgentà la section Référence des AWS CLI commandes.

`update-container-instances-state`

L'exemple de code suivant montre comment utiliser`update-container-instances-state`.

AWS CLI

Pour mettre à jour l'état d'une instance de conteneur

Ce qui suit `update-container-instances-state` met à jour l'état de l'instance de conteneur spécifiée dans `DRAINING` laquelle elle sera supprimée du cluster dans laquelle elle est enregistrée.

```
aws ecs update-container-instances-state \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \
    --status DRAINING
```

Sortie :

```
{
    "containerInstances": [
        {
            "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "ec2InstanceId": "i-013d87ffbb4d513bf",
            "version": 4390,
            "versionInfo": {
                "agentVersion": "1.29.0",
                "agentHash": "a190a73f",
                "dockerVersion": "DockerVersion:18.06.1-ce"
            },
            "remainingResources": [
                {
                    "name": "CPU",
```

```
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 1536
                },
                {
                    "name": "MEMORY",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 2681
                },
                {
                    "name": "PORTS",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678",
                        "51679"
                    ]
                },
                {
                    "name": "PORTS_UDP",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": []
                }
            ],
            "registeredResources": [
                {
                    "name": "CPU",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 2048
                },
                {
```

```
                    "name": "MEMORY",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 3705
                },
                {
                    "name": "PORTS",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678",
                        "51679"
                    ]
                },
                {
                    "name": "PORTS_UDP",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": []
                }
            ],
            "status": "DRAINING",
            "agentConnected": true,
            "runningTasksCount": 2,
            "pendingTasksCount": 0,
            "attributes": [
                {
                    "name": "ecs.capability.secrets.asm.environment-variables"
                },
                {
                    "name": "ecs.capability.branch-cni-plugin-version",
                    "value": "e0703516-"
                },
                {
                    "name": "ecs.ami-id",
                    "value": "ami-00e0090ac21971297"
```

```
                },
                {
                    "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.none"
                },
                {
                    "name": "ecs.capability.ecr-endpoint"
                },
                {
                    "name": "ecs.capability.docker-plugin.local"
                },
                {
                    "name": "ecs.capability.task-cpu-mem-limit"
                },
                {
                    "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
                },
                {
                    "name": "ecs.availability-zone",
                    "value": "us-west-2c"
                },
                {
                    "name": "ecs.capability.aws-appmesh"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
                },
                {
                    "name": "ecs.capability.task-eni-trunking"
                },
```

```
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
                },
                {
                    "name": "com.amazonaws.ecs.capability.privileged-container"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
                },
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                },
                {
                    "name": "com.amazonaws.ecs.capability.ecr-auth"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
                },
                {
                    "name": "ecs.os-type",
                    "value": "linux"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
                },
                {
                    "name": "ecs.capability.task-eia"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
                },
```

```
                {
                    "name": "ecs.capability.private-registry-
authentication.secretsmanager"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
                },
                {
                    "name": "ecs.capability.execution-role-awslogs"
                },
                {
                    "name": "ecs.vpc-id",
                    "value": "vpc-1234"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
                },
                {
                    "name": "ecs.capability.task-eni"
                },
                {
                    "name": "ecs.capability.execution-role-ecr-pull"
                },
                {
                    "name": "ecs.capability.container-health-check"
                },
                {
                    "name": "ecs.subnet-id",
                    "value": "subnet-1234"
                },
                {
                    "name": "ecs.instance-type",
                    "value": "c5.large"
                },
                {
```

```
                    "name": "com.amazonaws.ecs.capability.task-iam-role-network-
host"
                },
                {
                    "name": "ecs.capability.container-ordering"
                },
                {
                    "name": "ecs.capability.cni-plugin-version",
                    "value": "91ccefc8-2019.06.0"
                },
                {
                    "name": "ecs.capability.pid-ipc-namespace-sharing"
                },
                {
                    "name": "ecs.capability.secrets.ssm.environment-variables"
                },
                {
                    "name": "com.amazonaws.ecs.capability.task-iam-role"
                }
            ],
            "registeredAt": 1560788724.507,
            "attachments": [],
            "tags": []
        }
    ],
    "failures": []
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateContainerInstancesStateà la section Référence des AWS CLI commandes.

`update-service-primary-task-set`

L'exemple de code suivant montre comment utiliser`update-service-primary-task-set`.

AWS CLI

Pour mettre à jour l'ensemble de tâches principal d'un service

L'`update-service-primary-task-set`exemple suivant met à jour l'ensemble de tâches principal pour le service spécifié.

```
aws ecs update-service-primary-task-set \
```

```
    --cluster MyCluster \
    --service MyService \
    --primary-task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789
```

Sortie :

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
        "computedDesiredCount": 1,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557128360.711,
        "updatedAt": 1557129412.653,
        "launchType": "EC2",
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344312"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 50.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557129279.914
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateServicePrimaryTaskSetà la section Référence des AWS CLI commandes.

`update-service`

L'exemple de code suivant montre comment utiliser`update-service`.

AWS CLI

Exemple 1 : pour modifier la définition de tâche utilisée dans un service

L'`update-service`exemple suivant met à jour le `my-http-service` service pour utiliser la définition de `amazon-ecs-sample` tâche.

```
aws ecs update-service \
    --cluster test \
    --service my-http-service \
    --task-definition amazon-ecs-sample
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/test/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/test",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "1.4.0",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:2",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
```

```
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/7419115625193919142",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:2",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T13:26:02.734000-06:00",
                "updatedAt": "2025-02-21T13:26:02.734000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7419115625193919142 in
 progress."
            },
            {
                "id": "ecs-svc/1709597507655421668",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/old-amazon-ecs-sample:4",
```

```
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-01-24T11:13:07.621000-06:00",
                "updatedAt": "2025-02-02T16:11:30.838000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                             "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1709597507655421668
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "e40b4d1c-80d9-4834-aaf3-6a268e530e17",
                "createdAt": "2025-02-21T10:31:26.037000-06:00",
                "message": "(my-http-service) has reached a steady state."
            },
            {
                "id": "6ac069ad-fc8b-4e49-a35d-b5574a964c8e",
                "createdAt": "2025-02-21T04:31:22.703000-06:00",
                "message": "(my-http-service) has reached a steady state."
            },
            {
                "id": "265f7d37-dfd1-4880-a846-ec486f341919",
                "createdAt": "2025-02-20T22:31:22.514000-06:00",
                "message": "(my-http-service) has reached a steady state."
            }
        ],
```

```
        "createdAt": "2024-10-30T17:12:43.218000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321",
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "DISABLED"
    }
}
```

Pour plus d'informations, consultez Mettre à jour un service Amazon ECS à l'aide de la console dans le manuel Amazon ECS Developer Guide.

Exemple 2 : pour modifier le nombre de tâches dans un service

L'`update-service`exemple suivant met à jour le nombre de tâches souhaité du service `my-http-service` de 2 à 2.

```
aws ecs update-service \
    --cluster MyCluster \
    --service my-http-service \
    --desired-count 2
```

Sortie :

```
{
```

```
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 1,
        "pendingCount": 0,
        "capacityProviderStrategy": [
            {
                "capacityProvider": "FARGATE",
                "weight": 1,
                "base": 0
            }
        ],
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
MyTaskDefinition",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/1976744184940610707",
                "status": "PRIMARY",
                "taskkDefinition": "arn:aws:ecs:us-east-1:123456789012:task-
definition/MyTaskDefinition",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 1,
```

```
                "failedTasks": 0,
                "createdAt": "2024-12-03T16:24:25.225000-05:00",
                "updatedAt": "2024-12-03T16:25:15.837000-05:00",
                "capacityProviderStrategy": [
                    {
                        "capacityProvider": "FARGATE",
                        "weight": 1,
                        "base": 0
                    }
                ],
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-0d0eab1bb38d5ca64",
                            "subnet-0db5010045995c2d5"
                        ],
                        "securityGroups": [
                            "sg-02556bf85a191f59a"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1976744184940610707
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "f27350b9-4b2a-4e2e-b72e-a4b68380de45",
                "createdAt": "2024-12-30T13:24:07.345000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "e764ec63-f53f-45e3-9af2-d99f922d2957",
                "createdAt": "2024-12-30T12:32:21.600000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "28444756-c2fa-47f8-bd60-93a8e05f3991",
```

```
                "createdAt": "2024-12-08T19:26:10.367000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            }
        ],
        "createdAt": "2024-12-03T16:24:25.225000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-0d0eab1bb38d5ca64",
                    "subnet-0db5010045995c2d5"
                ],
                "securityGroups": [
                    "sg-02556bf85a191f59a"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/Admin",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un service Amazon ECS à l'aide de la console dans le manuel Amazon ECS Developer Guide.

Exemple 3 : pour configurer des volumes Amazon EBS pour les pièces jointes lors de la mise à jour du service

L'`update-service`exemple suivant met à jour le service `my-http-service` pour utiliser les volumes Amazon EBS. Vous devez avoir un rôle d'infrastructure Amazon ECS configuré avec la politique `AmazonECSInfrastructureRolePolicyForVolumes` gérée attachée. Vous devez également spécifier une définition de tâche portant le même nom de volume que celui indiqué

dans la `update-service` demande et `configuredAtLaunch` définie sur`true`. Cet exemple utilise l'`--cli-input-json`option et un fichier d'entrée JSON appelé`ebs.json`.

```
aws ecs update-service \
    --cli-input-json file://ebs.json
```

Contenu de `ebs.json` :

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "service": "my-http-service",
   "desiredCount": 2,
   "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
            }
        }
   ]
}
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
```

```
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:1",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/2420458347226626275",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:1",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:07:20.519000-06:00",
                "updatedAt": "2025-02-21T15:07:20.519000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321",
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
```

```
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/2420458347226626275 in
 progress.",
                "volumeConfigurations": [
                    {
                        "name": "ebs-volume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
                    }
                ]
            },
            {
                "id": "ecs-svc/5191625155316533644",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:2",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-02-21T14:54:48.862000-06:00",
                "updatedAt": "2025-02-21T14:57:22.502000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
```

```
                "rolloutStateReason": "ECS deployment ecs-svc/5191625155316533644
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "b5823113-c2c5-458e-9649-8c2ed38f23a5",
                "createdAt": "2025-02-21T14:57:22.508000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "b05a48e8-da35-4074-80aa-37ceb3167357",
                "createdAt": "2025-02-21T14:57:22.507000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/5191625155316533644) deployment completed."
            },
            {
                "id": "a10cd55d-4ba6-4cea-a655-5a5d32ada8a0",
                "createdAt": "2025-02-21T14:55:32.833000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 fb9c8df512684aec92f3c57dc3f22361)."
            },
        ],
        "createdAt": "2025-02-21T14:54:48.862000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
```

```
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

Pour plus d'informations, consultez la section Utiliser les volumes Amazon EBS avec Amazon ECS dans le manuel du développeur Amazon ECS.

Exemple 4 : pour mettre à jour un service afin de ne plus utiliser les volumes Amazon EBS

L'`update-service`exemple suivant met à jour le service `my-http-service` pour qu'il n'utilise plus les volumes Amazon EBS. Vous devez spécifier une révision de définition de tâche `configuredAtLaunch` définie sur`false`.

```
aws ecs update-service \
    --cluster mycluster \
    --task-definition mytaskdef \
    --service my-http-service \
    --desired-count 2 \
    --volume-configurations "[]"
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:3",
```

```
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/7522791612543716777",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:3",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:25:38.598000-06:00",
                "updatedAt": "2025-02-21T15:25:38.598000-06:00",
                    "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7522791612543716777 in
 progress."
            },
            {
```

```
                "id": "ecs-svc/2420458347226626275",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/myoldtaskdef:1",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:07:20.519000-06:00",
                "updatedAt": "2025-02-21T15:10:59.955000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/2420458347226626275
 completed.",
                "volumeConfigurations": [
                    {
                        "name": "ebs-volume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
                    }
                ]
            }
        ],
```

```
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "4f2c3ca1-7800-4048-ba57-bba210ada2ad",
                "createdAt": "2025-02-21T15:10:59.959000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "4b36a593-2d40-4ed6-8be8-b9b699eb6198",
                "createdAt": "2025-02-21T15:10:59.958000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/2420458347226626275) deployment completed."
            },
            {
                "id": "88380089-14e2-4ef0-8dbb-a33991683371",
                "createdAt": "2025-02-21T15:09:39.055000-06:00",
                "message": "(service my-http-service) has stopped 1 running tasks:
 (task fb9c8df512684aec92f3c57dc3f22361)."
            },
            {
                "id": "97d84243-d52f-4255-89bb-9311391c61f6",
                "createdAt": "2025-02-21T15:08:57.653000-06:00",
                "message": "(service my-http-service) has stopped 1 running tasks:
 (task 33eff090ad2c40539daa837e6503a9bc)."
            },
            {
                "id": "672ece6c-e2d0-4021-b5da-eefb14001687",
                "createdAt": "2025-02-21T15:08:15.631000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 996c02a66ff24f3190a4a8e0c841740f)."
            },
            {
                "id": "a3cf9bea-9be6-4175-ac28-4c68360986eb",
                "createdAt": "2025-02-21T15:07:36.931000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 d5d23c39f89e46cf9a647b9cc6572feb)."
            },
            {
                "id": "b5823113-c2c5-458e-9649-8c2ed38f23a5",
                "createdAt": "2025-02-21T14:57:22.508000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
```

```
                "id": "b05a48e8-da35-4074-80aa-37ceb3167357",
                "createdAt": "2025-02-21T14:57:22.507000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/5191625155316533644) deployment completed."
            },
            {
                "id": "a10cd55d-4ba6-4cea-a655-5a5d32ada8a0",
                "createdAt": "2025-02-21T14:55:32.833000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 fb9c8df512684aec92f3c57dc3f22361)."
            },
            {
                "id": "42da91fa-e26d-42ef-88c3-bb5965c56b2f",
                "createdAt": "2025-02-21T14:55:02.703000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 33eff090ad2c40539daa837e6503a9bc)."
            }
        ],
        "createdAt": "2025-02-21T14:54:48.862000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
```

```
}
```

Pour plus d'informations, consultez la section Utiliser les volumes Amazon EBS avec Amazon ECS dans le manuel du développeur Amazon ECS.

Exemple 5 : pour activer le rééquilibrage des zones de disponibilité pour un service

L'`update-service`exemple suivant active le rééquilibrage des zones de disponibilité pour le service`my-http-service`.

```
aws ecs update-service \
    --cluster MyCluster \
    --service my-http-service \
    --availability-zone-rebalancing ENABLED
```

Sortie :

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 1,
        "pendingCount": 0,
        "capacityProviderStrategy": [
            {
                "capacityProvider": "FARGATE",
                "weight": 1,
                "base": 0
            }
        ],
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
MyTaskDefinition",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
```

```
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/1976744184940610707",
                "status": "PRIMARY",
                "taskkDefinition": "arn:aws:ecs:us-east-1:123456789012:task-
definition/MyTaskDefinition",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 1,
                "failedTasks": 0,
                "createdAt": "2024-12-03T16:24:25.225000-05:00",
                "updatedAt": "2024-12-03T16:25:15.837000-05:00",
                "capacityProviderStrategy": [
                    {
                        "capacityProvider": "FARGATE",
                        "weight": 1,
                        "base": 0
                    }
                ],
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-0d0eab1bb38d5ca64",
                            "subnet-0db5010045995c2d5"
                        ],
                        "securityGroups": [
                            "sg-02556bf85a191f59a"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
```

```
                "rolloutStateReason": "ECS deployment ecs-svc/1976744184940610707
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": "2024-12-03T16:24:25.225000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-0d0eab1bb38d5ca64",
                    "subnet-0db5010045995c2d5"
                ],
                "securityGroups": [
                    "sg-02556bf85a191f59a"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/Admin",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un service Amazon ECS à l'aide de la console dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceà la section Référence des AWS CLI commandes.

`update-task-protection`

L'exemple de code suivant montre comment utiliser`update-task-protection`.

AWS CLI

Exemple 1 : activer la protection des tâches pour les tâches ECS

Ce qui suit `update-task-protection` protège votre tâche ECS contre toute interruption lors de la mise en place à partir de déploiements ou de services. AutoScaling Vous pouvez définir une période d'expiration personnalisée pour la protection des tâches comprise entre 1 et 2 880 minutes (48 heures). Si vous ne spécifiez pas de période d'expiration, la durée par défaut d'activation de la protection des tâches est de 2 heures.

```
aws ecs update-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24 \
    --protection-enabled \
    --expires-in-minutes 300
```

Sortie :

```
{
"protectedTasks": [
    {
        "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
        "protectionEnabled": true,
        "expirationDate": "2024-09-14T19:53:36.687000-05:00"
    }
],
"failures": []
}
```

Exemple 2 : désactiver la protection des tâches pour les tâches ECS

Ce qui suit `update-task-protection` désactive les tâches protégées contre l'évolutivité provenant des déploiements ou des services. AutoScaling

```
aws ecs update-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24 \
```

```
    --no-protection-enabled
```

Sortie :

```
{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

Pour en savoir plus sur la protection des tâches, consultez la section Protéger vos tâches Amazon ECS contre l'interruption à la suite d'événements évolutifs dans le manuel Amazon ECS Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateTaskProtectionà la section Référence des AWS CLI commandes.

`update-task-set`

L'exemple de code suivant montre comment utiliser`update-task-set`.

AWS CLI

Pour mettre à jour un ensemble de tâches

L'`update-task-set`exemple suivant met à jour un ensemble de tâches pour ajuster l'échelle.

```
aws ecs update-task-set \
    --cluster MyCluster \
    --service MyService \
    --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-
svc/1234567890123456789 \
    --scale value=50,unit=PERCENT
```

Sortie :

```
{
```

```
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
        "computedDesiredCount": 0,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557128360.711,
        "updatedAt": 1557129279.914,
        "launchType": "EC2",
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 50.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557129279.914
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateTaskSetà la section Référence des AWS CLI commandes.

Exemples d'Amazon EFS utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon EFS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-file-system`

L'exemple de code suivant montre comment utiliser`create-file-system`.

AWS CLI

Pour créer un système de fichiers chiffré

L'`create-file-system`exemple suivant crée un système de fichiers chiffré à l'aide de la clé CMK par défaut. Il ajoute également le tag`Name=my-file-system`.

```
aws efs create-file-system \
    --performance-mode generalPurpose \
    --throughput-mode bursting \
    --encrypted \
    --tags Key=Name,Value=my-file-system
```

Sortie :

```
{
    "OwnerId": "123456789012",
    "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
    "FileSystemId": "fs-c7a0456e",
    "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/
fs-48499b4d",
    "CreationTime": 1595286880.0,
    "LifeCycleState": "creating",
    "Name": "my-file-system",
    "NumberOfMountTargets": 0,
    "SizeInBytes": {
```

```
        "Value": 0,
        "ValueInIA": 0,
        "ValueInStandard": 0
    },
    "PerformanceMode": "generalPurpose",
    "Encrypted": true,
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-
adcf-30d92example",
    "ThroughputMode": "bursting",
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création de systèmes de fichiers Amazon EFS dans le guide de l'utilisateur Amazon Elastic File System.

- Pour plus de détails sur l'API, reportez-vous CreateFileSystemà la section Référence des AWS CLI commandes.

`create-mount-target`

L'exemple de code suivant montre comment utiliser`create-mount-target`.

AWS CLI

Pour créer une cible de montage

L'`create-mount-target`exemple suivant crée une cible de montage pour le système de fichiers spécifié.

```
aws efs create-mount-target \
    --file-system-id fs-c7a0456e \
    --subnet-id subnet-02bf4c428bexample \
    --security-groups sg-068f739363example
```

Sortie :

```
{
```

```
    "OwnerId": "123456789012",
    "MountTargetId": "fsmt-f9a14450",
    "FileSystemId": "fs-c7a0456e",
    "SubnetId": "subnet-02bf4c428bexample",
    "LifeCycleState": "creating",
    "IpAddress": "10.0.1.24",
    "NetworkInterfaceId": "eni-02d542216aexample",
    "AvailabilityZoneId": "use2-az2",
    "AvailabilityZoneName": "us-east-2b",
    "VpcId": "vpc-0123456789abcdef0"
}
```

Pour plus d'informations, consultez la section Création de cibles de montage dans le guide de l'utilisateur d'Amazon Elastic File System.

- Pour plus de détails sur l'API, reportez-vous CreateMountTargetà la section Référence des AWS CLI commandes.

`delete-file-system`

L'exemple de code suivant montre comment utiliser`delete-file-system`.

AWS CLI

Pour supprimer un système de fichiers

L'`delete-file-system`exemple suivant supprime le système de fichiers spécifié.

```
aws efs delete-file-system \
    --file-system-id fs-c7a0456e
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un système de fichiers Amazon EFS dans le guide de l'utilisateur Amazon Elastic File System.

- Pour plus de détails sur l'API, reportez-vous DeleteFileSystemà la section Référence des AWS CLI commandes.

`delete-mount-target`

L'exemple de code suivant montre comment utiliser`delete-mount-target`.

AWS CLI

Pour supprimer une cible de montage

L'`delete-mount-target`exemple suivant supprime la cible de montage spécifiée.

```
aws efs delete-mount-target \
    --mount-target-id fsmt-f9a14450
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création de cibles de montage dans le guide de l'utilisateur d'Amazon Elastic File System.

- Pour plus de détails sur l'API, reportez-vous DeleteMountTargetà la section Référence des AWS CLI commandes.

`describe-file-systems`

L'exemple de code suivant montre comment utiliser`describe-file-systems`.

AWS CLI

Pour décrire un système de fichiers

L'`describe-file-systems`exemple suivant décrit le système de fichiers spécifié.

```
aws efs describe-file-systems \
    --file-system-id fs-c7a0456e
```

Sortie :

```
{
    "FileSystems": [
        {
            "OwnerId": "123456789012",
            "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
            "FileSystemId": "fs-c7a0456e",
            "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-48499b4d",
            "CreationTime": 1595286880.0,
            "LifeCycleState": "available",
```

```
            "Name": "my-file-system",
            "NumberOfMountTargets": 3,
            "SizeInBytes": {
                "Value": 6144,
                "Timestamp": 1600991437.0,
                "ValueInIA": 0,
                "ValueInStandard": 6144
            },
            "PerformanceMode": "generalPurpose",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-
adcf-30d92example",
            "ThroughputMode": "bursting",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-file-system"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des systèmes de fichiers Amazon EFS dans le guide de l'utilisateur Amazon Elastic File System.

- Pour plus de détails sur l'API, reportez-vous DescribeFileSystemsà la section Référence des AWS CLI commandes.

`describe-mount-targets`

L'exemple de code suivant montre comment utiliser`describe-mount-targets`.

AWS CLI

Pour décrire une cible de montage

L'`describe-mount-targets`exemple suivant décrit la cible de montage spécifiée.

```
aws efs describe-mount-targets \
    --mount-target-id fsmt-f9a14450
```

Sortie :

```
{
    "MountTargets": [
        {
            "OwnerId": "123456789012",
            "MountTargetId": "fsmt-f9a14450",
            "FileSystemId": "fs-c7a0456e",
            "SubnetId": "subnet-02bf4c428bexample",
            "LifeCycleState": "creating",
            "IpAddress": "10.0.1.24",
            "NetworkInterfaceId": "eni-02d542216aexample",
            "AvailabilityZoneId": "use2-az2",
            "AvailabilityZoneName": "us-east-2b",
            "VpcId": "vpc-0123456789abcdef0"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création de cibles de montage dans le guide de l'utilisateur d'Amazon Elastic File System.

- Pour plus de détails sur l'API, reportez-vous DescribeMountTargetsà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire les balises d'un système de fichiers

L'`describe-tags`exemple suivant décrit les balises du système de fichiers spécifié.

```
aws efs describe-tags \
    --file-system-id fs-c7a0456e
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Name",
```

```
            "Value": "my-file-system"
        },
        {
            "Key": "Department",
            "Value": "Business Intelligence"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des balises du système de fichiers dans le manuel Amazon Elastic File System User Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour récupérer les balises d'une ressource

L'`list-tags-for-resource`exemple suivant récupère les balises associées au système de fichiers spécifié.

```
aws efs list-tags-for-resource \
    --resource-id fs-c7a0456e
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        },
        {
            "Key": "Department",
            "Value": "Business Intelligence"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Gestion des balises du système de fichiers dans le manuel Amazon Elastic File System User Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute la balise `Department=Business Intelligence` au système de fichiers spécifié.

```
aws efs tag-resource \
    --resource-id fs-c7a0456e \
    --tags Key=Department,Value="Business Intelligence"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises du système de fichiers dans le manuel Amazon Elastic File System User Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime la balise avec la clé de `Department` balise du système de fichiers spécifié.

```
aws efs untag-resource \
    --resource-id fs-c7a0456e \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises du système de fichiers dans le manuel Amazon Elastic File System User Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon EKS AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon EKS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-access-policy`

L'exemple de code suivant montre comment utiliser`associate-access-policy`.

AWS CLI

Pour associer une politique d'accès et son étendue à l'entrée d'accès du cluster

Ce qui suit `associate-access-policy` associe une politique d'accès et son étendue à l'entrée d'accès du cluster spécifié.

```
aws eks associate-access-policy \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy \
    --access-scope type=namespace,namespaces=default
```

Sortie :

```
{
    "clusterName": "eks-customer",
    "principalArn": "arn:aws:iam::111122223333:role/Admin",
    "associatedAccessPolicy": {
        "policyArn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy",
        "accessScope": {
            "type": "namespace",
            "namespaces": [
                "default"
            ]
        },
        "associatedAt": "2025-05-24T15:59:51.981000-05:00",
        "modifiedAt": "2025-05-24T15:59:51.981000-05:00"
    }
}
```

Pour plus d'informations, consultez Associer les politiques d'accès aux entrées d'accès dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir AssociateAccessPolicyla section Référence des AWS CLI commandes.

`associate-encryption-config`

L'exemple de code suivant montre comment utiliser`associate-encryption-config`.

AWS CLI

Pour associer une configuration de chiffrement à un cluster existant

L'`associate-encryption-config`exemple suivant active le chiffrement sur un cluster EKS existant sur lequel le chiffrement n'est pas encore activé.

```
aws eks associate-encryption-config \
```

```
    --cluster-name my-eks-cluster \
    --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

Sortie :

```
{
    "update": {
        "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
        "status": "InProgress",
        "type": "AssociateEncryptionConfig",
        "params": [
            {
                "type": "EncryptionConfig",
                "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
            }
        ],
        "createdAt": "2024-03-14T11:01:26.297000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Activation du chiffrement secret sur un cluster existant dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir AssociateEncryptionConfigla section Référence des AWS CLI commandes.

`associate-identity-provider-config`

L'exemple de code suivant montre comment utiliser`associate-identity-provider-config`.

AWS CLI

Associez un fournisseur d'identité à votre cluster Amazon EKS

L'`associate-identity-provider-config`exemple suivant associe un fournisseur d'identité à votre cluster Amazon EKS.

```
aws eks associate-identity-provider-config \
    --cluster-name my-eks-cluster \
```

```
    --oidc 'identityProviderConfigName=my-identity-provider,issuerUrl=https://
oidc.eks.us-east-2.amazonaws.com/
id/38D6A4619A0A69E342B113ED7F1A7652,clientId=kubernetes,usernameClaim=email,usernamePrefix=m
username-prefix,groupsClaim=my-claim,groupsPrefix=my-groups-
prefix,requiredClaims={Claim1=value1,Claim2=value2}' \
    --tags env=dev
```

Sortie :

```
{
    "update": {
        "id": "8c6c1bef-61fe-42ac-a242-89412387b8e7",
        "status": "InProgress",
        "type": "AssociateIdentityProviderConfig",
        "params": [
            {
                "type": "IdentityProviderConfig",
                "value": "[{\"type\":\"oidc\",\"name\":\"my-identity-provider\"}]"
            }
        ],
        "createdAt": "2024-04-11T13:46:49.648000-04:00",
        "errors": []
    },
    "tags": {
        "env": "dev"
    }
}
```

Pour plus d'informations, consultez Authentifier les utilisateurs de votre cluster auprès d'un fournisseur d'identité OpenID Connect - Associer un fournisseur d'identité OIDC dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir AssociateIdentityProviderConfigla section Référence des AWS CLI commandes.

`create-access-entry`

L'exemple de code suivant montre comment utiliser`create-access-entry`.

AWS CLI

Exemple 1 : pour créer l'entrée d'accès pour le cluster EKS

L'`create-access-entry`exemple suivant crée une entrée d'accès qui permet à un principal IAM d'accéder au cluster EKS.

```
aws eks create-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:user/eks-user
```

Sortie :

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:user/eks-user",
        "kubernetesGroups": [],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/user/111122223333/eks-user/a1b2c3d4-5678-90ab-cdef-a6506e3d36p0",
        "createdAt": "2025-04-14T22:45:48.097000-05:00",
        "modifiedAt": "2025-04-14T22:45:48.097000-05:00",
        "tags": {},
        "username": "arn:aws:iam::111122223333:user/eks-user",
        "type": "STANDARD"
    }
}
```

Pour plus d'informations, consultez la section Créer des entrées d'accès dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : pour créer l'entrée d'accès pour le cluster EKS en spécifiant le type d'entrée d'accès

L'`create-access-entry`exemple suivant crée une entrée d'accès de type `EC2_LINUX` dans le cluster EKS. Par défaut, une entrée d'`STANDARD`accès de type est créée. Outre la valeur par défaut, si nous spécifiez d'autres types d'entrée d'accès, un ARN de rôle IAM doit être transmis dans la CLI.

```
aws eks create-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/admin-test-ip \
    --type EC2_LINUX
```

Sortie :

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:role/admin-test-ip",
        "kubernetesGroups": [
            "system:nodes"
        ],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/role/111122223333/admin-test-ip/accb5418-f493-f390-3e6e-c3f19f725fcp",
        "createdAt": "2025-05-06T19:42:45.453000-05:00",
        "modifiedAt": "2025-05-06T19:42:45.453000-05:00",
        "tags": {},
        "username": "system:node:{{EC2PrivateDNSName}}",
        "type": "EC2_LINUX"
    }
}
```

Pour plus d'informations, consultez la section Créer des entrées d'accès dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir CreateAccessEntryla section Référence des AWS CLI commandes.

`create-addon`

L'exemple de code suivant montre comment utiliser`create-addon`.

AWS CLI

Exemple 1 : pour créer un module complémentaire Amazon EKS avec une version compatible par défaut pour la version du cluster EKS correspondante

L'`create-addon`exemple de commande suivant crée un module complémentaire Amazon EKS avec une version compatible par défaut pour la version du cluster EKS correspondante.

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.15.1-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/1ec71ee1-b9c2-8915-4e17-e8be0a55a149",
        "createdAt": "2024-03-14T12:20:03.264000-04:00",
        "modifiedAt": "2024-03-14T12:20:03.283000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : pour créer un module complémentaire Amazon EKS avec une version complémentaire spécifique

L'`create-addon`exemple de commande suivant crée un module complémentaire Amazon EKS avec une version spécifique du module complémentaire.

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
```

```
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/34c71ee6-7738-6c8b-c6bd-3921a176b5ff",
        "createdAt": "2024-03-14T12:30:24.507000-04:00",
        "modifiedAt": "2024-03-14T12:30:24.521000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : pour créer un module complémentaire Amazon EKS avec des valeurs de configuration personnalisées et résoudre les détails des conflits

L'`create-addon`exemple de commande suivant crée un module complémentaire Amazon EKS avec des valeurs de configuration personnalisées et résout les détails des conflits.

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values '{"resources":{"limits":{"cpu":"100m"}}}' \
    --resolve-conflicts OVERWRITE
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/a6c71ee9-0304-9237-1be8-25af1b0f1ffb",
        "createdAt": "2024-03-14T12:35:58.313000-04:00",
        "modifiedAt": "2024-03-14T12:35:58.327000-04:00",
```

```
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {},
        "configurationValues": "{\"resources\":{\"limits\":{\"cpu\":\"100m\"}}}"
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 4 : pour créer un module complémentaire Amazon EKS avec un fichier de valeurs de configuration JSON personnalisé

L'`create-addon`exemple de commande suivant crée un module complémentaire Amazon EKS avec des valeurs de configuration personnalisées et des détails de résolution des conflits.

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values 'file://configuration-values.json' \
    --resolve-conflicts OVERWRITE \
    --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

Contenu de `configuration-values.json` :

```
{
    "resources": {
        "limits": {
            "cpu": "150m"
        }
    },
    "env": {
        "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
    }
}
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
```

```
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/d8c71ef8-fbd8-07d0-fb32-6a7be19ececd",
        "createdAt": "2024-03-14T13:10:51.763000-04:00",
        "modifiedAt": "2024-03-14T13:10:51.777000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {
            "eks-addon-key-1": "value-1",
            "eks-addon-key-2": "value-2"
        },
        "configurationValues": "{\n    \"resources\": {\n        \"limits\":
 {\n            \"cpu\": \"150m\"\n        }\n    },\n    \"env\": {\n
 \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n    }\n}"
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 5 : pour créer un module complémentaire Amazon EKS avec un fichier de valeurs de configuration YAML personnalisé

L'`create-addon`exemple de commande suivant crée un module complémentaire Amazon EKS avec des valeurs de configuration personnalisées et des détails de résolution des conflits.

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values 'file://configuration-values.yaml' \
    --resolve-conflicts OVERWRITE \
    --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

Contenu de `configuration-values.yaml` :

```
resources:
```

```
    limits:
        cpu: '100m'
env:
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/d4c71efb-3909-6f36-a548-402cd4b5d59e",
        "createdAt": "2024-03-14T13:15:45.220000-04:00",
        "modifiedAt": "2024-03-14T13:15:45.237000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {
            "eks-addon-key-3": "value-3",
            "eks-addon-key-4": "value-4"
        },
        "configurationValues": "resources:\n    limits:\n        cpu: '100m'\nenv:\n
    AWS_VPC_K8S_CNI_LOGLEVEL: 'INFO'"
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir CreateAddonla section Référence des AWS CLI commandes.

`create-cluster`

L'exemple de code suivant montre comment utiliser`create-cluster`.

AWS CLI

Pour créer un nouveau cluster

Cet exemple de commande crée un cluster nommé `prod` dans votre région par défaut.

Commande :

```
aws eks create-cluster --name prod \
--role-arn arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J7ONKE3BQ4PI \
--resources-vpc-config subnetIds=subnet-6782e71e,subnet-
e7e761ac,securityGroupIds=sg-6979fe18
```

Sortie :

```
{
    "cluster": {
        "name": "prod",
        "arn": "arn:aws:eks:us-west-2:012345678910:cluster/prod",
        "createdAt": 1527808069.147,
        "version": "1.10",
        "roleArn": "arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J7ONKE3BQ4PI",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-6782e71e",
                "subnet-e7e761ac"
            ],
            "securityGroupIds": [
                "sg-6979fe18"
            ],
            "vpcId": "vpc-950809ec"
        },
        "status": "CREATING",
        "certificateAuthority": {}
    }
}
```

Pour créer un nouveau cluster avec l'accès aux terminaux privés et la journalisation activés

Cet exemple de commande crée un cluster nommé `example` dans votre région par défaut avec l'accès au point de terminaison public désactivé, l'accès au point de terminaison privé activé et tous les types de journalisation activés.

Commande :

```
aws eks create-cluster --name example --kubernetes-version 1.12 \
--role-arn arn:aws:iam::012345678910:role/example-cluster-ServiceRole-1XWBQWYSFRE2Q
 \
--resources-vpc-
config subnetIds=subnet-0a188dccd2f9a632f,subnet-09290d93da4278664,subnet-0f21dd86e0e91134a,
 \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Sortie :

```
{
    "cluster": {
        "name": "example",
        "arn": "arn:aws:eks:us-west-2:012345678910:cluster/example",
        "createdAt": 1565804921.901,
        "version": "1.12",
        "roleArn": "arn:aws:iam::012345678910:role/example-cluster-
ServiceRole-1XWBQWYSFRE2Q",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0a188dccd2f9a632f",
                "subnet-09290d93da4278664",
                "subnet-0f21dd86e0e91134a",
                "subnet-0173dead68481a583",
                "subnet-051f70a57ed6fcab6",
                "subnet-01322339c5c7de9b4"
            ],
            "securityGroupIds": [
                "sg-0c5b580845a031c10"
            ],
            "vpcId": "vpc-0f622c01f68d4afec",
            "endpointPublicAccess": false,
            "endpointPrivateAccess": true
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
```

```
                        "scheduler"
                    ],
                    "enabled": true
                }
            ]
        },
        "status": "CREATING",
        "certificateAuthority": {},
        "platformVersion": "eks.3"
    }
}
```

- Pour plus de détails sur l'API, voir CreateClusterla section Référence des AWS CLI commandes.

`create-fargate-profile`

L'exemple de code suivant montre comment utiliser`create-fargate-profile`.

AWS CLI

Exemple 1 : créer un profil EKS Fargate pour un sélecteur avec un espace de noms

L'`create-fargate-profile`exemple suivant crée un profil Fargate EKS pour un sélecteur avec un espace de noms.

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default"}]'
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/a2c72bca-318e-abe8-8ed1-27c6d4892e9e",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:38:47.368000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
```

```
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default"
            }
        ],
        "status": "CREATING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Profil AWS Fargate - Création d'un profil Fargate dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : créer un profil EKS Fargate pour un sélecteur avec un espace de noms et des étiquettes

L'`create-fargate-profile`exemple suivant crée un profil Fargate EKS pour un sélecteur avec un espace de noms et des étiquettes.

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default", "labels": {"labelname1":
 "labelvalue1"}}]'
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/88c72bc7-e8a4-fa34-44e4-2f1397224bb3",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:33:48.125000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
```

```
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default",
                "labels": {
                    "labelname1": "labelvalue1"
                }
            }
        ],
        "status": "CREATING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Profil AWS Fargate - Création d'un profil Fargate dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : créer un profil EKS Fargate pour un sélecteur avec un espace de noms et des étiquettes, ainsi que des sous-réseaux dans lesquels IDs lancer un pod.

L'`create-fargate-profile`exemple suivant crée le profil EKS Fargate pour un sélecteur avec un espace de noms et des étiquettes, ainsi que IDs des sous-réseaux dans lesquels lancer un Pod.

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default", "labels": {"labelname1":
 "labelvalue1"}}]' \
    --subnets '["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
 "subnet-0e2907431c9988b72"]'
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/e8c72bc8-e87b-5eb6-57cb-ed4fe57577e3",
```

```
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:35:58.640000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default",
                "labels": {
                    "labelname1": "labelvalue1"
                }
            }
        ],
        "status": "CREATING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Profil AWS Fargate - Création d'un profil Fargate dans le guide de l'utilisateur Amazon EKS.

Exemple 4 : créer un profil EKS Fargate pour un sélecteur avec plusieurs espaces de noms et étiquettes, ainsi que des sous-réseaux dans lesquels lancer IDs un pod

L'`create-fargate-profile`exemple suivant crée un profil Fargate EKS pour un sélecteur avec plusieurs espaces de noms et étiquettes, ainsi que des sous-réseaux dans lesquels lancer un IDs Pod.

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default1", "labels": {"labelname1": "labelvalue1",
 "labelname2": "labelvalue2"}}, {"namespace": "default2", "labels": {"labelname1":
 "labelvalue1", "labelname2": "labelvalue2"}}]' \
    --subnets '["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
 "subnet-0e2907431c9988b72"]' \
    --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2":
 "value-2"}'
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/4cc72bbf-b766-8ee6-8d29-e62748feb3cd",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:15:55.271000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default1",
                "labels": {
                    "labelname2": "labelvalue2",
                    "labelname1": "labelvalue1"
                }
            },
            {
                "namespace": "default2",
                "labels": {
                    "labelname2": "labelvalue2",
                    "labelname1": "labelvalue1"
                }
            }
        ],
        "status": "CREATING",
        "tags": {
            "eks-fargate-profile-key-2": "value-2",
            "eks-fargate-profile-key-1": "value-1"
        }
    }
}
```

Pour plus d'informations, consultez la section Profil AWS Fargate - Création d'un profil Fargate dans le guide de l'utilisateur Amazon EKS.

Exemple 5 : Création d'un profil EKS Fargate avec un sélecteur générique pour les espaces de noms et les étiquettes, ainsi que des sous-réseaux dans lesquels lancer IDs un pod

L'`create-fargate-profile`exemple suivant crée un profil Fargate EKS pour un sélecteur avec plusieurs espaces de noms et étiquettes, ainsi que des sous-réseaux dans lesquels lancer un IDs Pod.

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "prod*", "labels": {"labelname*?": "*value1"}},
 {"namespace": "*dev*", "labels": {"labelname*?": "*value*"}}]' \
    --subnets '["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
 "subnet-0e2907431c9988b72"]' \
    --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2":
 "value-2"}'
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/e8c72bd6-5966-0bfe-b77b-1802893e5a6f",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T13:05:20.550000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "prod*",
                "labels": {
                    "labelname*?": "*value1"
                }
            },
            {
                "namespace": "*dev*",
                "labels": {
```

```
                    "labelname*?": "*value*"
                }
            }
        ],
        "status": "CREATING",
        "tags": {
            "eks-fargate-profile-key-2": "value-2",
            "eks-fargate-profile-key-1": "value-1"
        }
    }
}
```

Pour plus d'informations, consultez la section Profil AWS Fargate - Création d'un profil Fargate dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir CreateFargateProfilela section Référence des AWS CLI commandes.

`create-nodegroup`

L'exemple de code suivant montre comment utiliser`create-nodegroup`.

AWS CLI

Exemple 1 : crée un groupe de nœuds gérés pour un cluster Amazon EKS

L'`create-nodegroup`exemple suivant crée un groupe de nœuds gérés pour un cluster Amazon EKS.

```
aws eks create-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --node-role arn:aws:iam::111122223333:role/role-name \
    --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
 \
    --scaling-config minSize=1,maxSize=3,desiredSize=1 \
    --region us-east-2
```

Sortie :

```
{
    "nodegroup": {
```

```
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/bac7550f-b8b8-5fbb-4f3e-7502a931119e",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-04T13:19:32.260000-04:00",
        "modifiedAt": "2024-04-04T13:19:32.260000-04:00",
        "status": "CREATING",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 3,
            "desiredSize": 1
        },
        "instanceTypes": [
            "t3.medium"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72, subnet-04ad87f71c6e5ab4d,
 subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "diskSize": 20,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : crée un groupe de nœuds gérés pour un cluster Amazon EKS avec des types d'instances et une taille de disque personnalisés

L'`create-nodegroup`exemple suivant crée un groupe de nœuds gérés pour un cluster Amazon EKS avec des types d'instances et une taille de disque personnalisés.

```
aws eks create-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --node-role arn:aws:iam::111122223333:role/role-name \
    --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
 \
    --scaling-config minSize=1,maxSize=3,desiredSize=1 \
    --capacity-type ON_DEMAND \
    --instance-types 'm5.large' \
    --disk-size 50 \
    --region us-east-2
```

Sortie :

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/c0c7551b-e4f9-73d9-992c-a450fdb82322",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-04T13:46:07.595000-04:00",
        "modifiedAt": "2024-04-04T13:46:07.595000-04:00",
        "status": "CREATING",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 3,
            "desiredSize": 1
        },
        "instanceTypes": [
            "m5.large"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "diskSize": 50,
```

```
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : crée un groupe de nœuds gérés pour un cluster Amazon EKS avec des types d'instances, une taille de disque, un type ami, un type de capacité, une configuration de mise à jour, des étiquettes, des marques et des balises personnalisés.

L'`create-nodegroup`exemple suivant crée un groupe de nœuds gérés pour un cluster Amazon EKS avec des types d'instances, une taille de disque, un type ami, un type de capacité, une configuration de mise à jour, des étiquettes, des marques et des balises personnalisés.

```
aws eks create-nodegroup  \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --node-role arn:aws:iam::111122223333:role/role-name \
    --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
 \
    --scaling-config minSize=1,maxSize=5,desiredSize=4 \
    --instance-types 't3.large' \
    --disk-size 50 \
    --ami-type AL2_x86_64 \
    --capacity-type SPOT \
    --update-config maxUnavailable=2 \
    --labels '{"my-eks-nodegroup-label-1": "value-1" , "my-eks-nodegroup-label-2":
 "value-2"}' \
    --taints '{"key": "taint-key-1" , "value": "taint-value-1", "effect":
 "NO_EXECUTE"}' \
    --tags '{"my-eks-nodegroup-key-1": "value-1" , "my-eks-nodegroup-key-2":
 "value-2"}'
```

Sortie :

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/88c75524-97af-0cb9-a9c5-7c0423ab5314",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-04T14:05:07.940000-04:00",
        "modifiedAt": "2024-04-04T14:05:07.940000-04:00",
        "status": "CREATING",
        "capacityType": "SPOT",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 5,
            "desiredSize": 4
        },
        "instanceTypes": [
            "t3.large"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "labels": {
            "my-eks-nodegroup-label-2": "value-2",
            "my-eks-nodegroup-label-1": "value-1"
        },
        "taints": [
            {
                "key": "taint-key-1",
                "value": "taint-value-1",
                "effect": "NO_EXECUTE"
            }
        ],
        "diskSize": 50,
        "health": {
            "issues": []
        },
        "updateConfig": {
```

```
            "maxUnavailable": 2
        },
        "tags": {
            "my-eks-nodegroup-key-1": "value-1",
            "my-eks-nodegroup-key-2": "value-2"
        }
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir CreateNodegroupla section Référence des AWS CLI commandes.

`create-pod-identity-association`

L'exemple de code suivant montre comment utiliser`create-pod-identity-association`.

AWS CLI

Exemple 1 : pour créer une association EKS Pod Identity dans un cluster EKS

L'`create-pod-identity-association`exemple suivant crée une association EKS Pod Identity entre un compte de service du cluster EKS et un rôle IAM.

```
aws eks create-pod-identity-association \
    --cluster-name eks-customer \
    --namespace default \
    --service-account default \
    --role-arn arn:aws:iam::111122223333:role/my-role
```

Sortie :

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-8mvwvh57cu74mgcst",
```

```
        "associationId": "a-8mvwvh57cu74mgcst",
        "tags": {},
        "createdAt": "2025-05-24T19:40:13.961000-05:00",
        "modifiedAt": "2025-05-24T19:40:13.961000-05:00"
    }
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : pour créer une association EKS Pod Identity dans un cluster EKS avec des balises

Ce qui suit `create-pod-identity-association` crée une association EKS Pod Identity entre un compte de service et un rôle IAM dans le cluster EKS avec des balises.

```
aws eks create-pod-identity-association \
    --cluster-name eks-customer \
    --namespace default \
    --service-account default \
    --role-arn arn:aws:iam::111122223333:role/my-role \
    --tags Key1=value1,Key2=value2
```

Sortie :

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgoda",
        "associationId": "a-9njjin9gfghecgoda",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-24T19:52:14.135000-05:00"
    }
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir CreatePodIdentityAssociationla section Référence des AWS CLI commandes.

`delete-access-entry`

L'exemple de code suivant montre comment utiliser`delete-access-entry`.

AWS CLI

Pour supprimer une entrée d'accès associée au cluster

Ce qui suit `delete-access-entry` supprime une entrée d'accès associée au cluster EKS nommé`eks-customer`.

```
aws eks delete-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer les entrées d'accès dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DeleteAccessEntryla section Référence des AWS CLI commandes.

`delete-addon`

L'exemple de code suivant montre comment utiliser`delete-addon`.

AWS CLI

Exemple 1. Pour supprimer un module complémentaire Amazon EKS tout en préservant le logiciel complémentaire sur le cluster EKS

L'`delete-addon`exemple de commande suivant supprime un module complémentaire Amazon EKS tout en préservant le logiciel complémentaire sur le cluster EKS.

```
aws eks delete-addon \
```

```
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --preserve
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "DELETING",
        "addonVersion": "v1.9.3-eksbuild.7",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/a8c71ed3-944e-898b-9167-c763856af4b8",
        "createdAt": "2024-03-14T11:49:09.009000-04:00",
        "modifiedAt": "2024-03-14T12:03:49.776000-04:00",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez Gérer les modules complémentaires Amazon EKS - Supprimer un module complémentaire dans Amazon EKS.

Exemple 2. Pour supprimer un module complémentaire Amazon EKS et également supprimer le logiciel complémentaire du cluster EKS

L'`delete-addon`exemple de commande suivant supprime un module complémentaire Amazon EKS et supprime également le logiciel complémentaire du cluster EKS.

```
aws eks delete-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon
```

Sortie :

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
```

```
        "status": "DELETING",
        "addonVersion": "v1.15.1-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/bac71ed1-ec43-3bb6-88ea-f243cdb58954",
        "createdAt": "2024-03-14T11:45:31.983000-04:00",
        "modifiedAt": "2024-03-14T11:58:40.136000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez Gérer les modules complémentaires Amazon EKS - Supprimer un module complémentaire dans Amazon EKS.

- Pour plus de détails sur l'API, voir DeleteAddonla section Référence des AWS CLI commandes.

`delete-cluster`

L'exemple de code suivant montre comment utiliser`delete-cluster`.

AWS CLI

Supprimer un plan de contrôle de cluster Amazon EKS

L'`delete-cluster`exemple suivant supprime un plan de contrôle de cluster Amazon EKS.

```
aws eks delete-cluster \
    --name my-eks-cluster
```

Sortie :

```
{
    "cluster": {
        "name": "my-eks-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
        "createdAt": "2024-03-14T11:31:44.348000-04:00",
        "version": "1.27",
        "endpoint": "https://DALSJ343KE23J3RN45653DSKJTT647TYD.yl4.us-
east-2.eks.amazonaws.com",
```

```
        "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0fb75d2d8401716e7",
                "subnet-02184492f67a3d0f9",
                "subnet-04098063527aab776",
                "subnet-0e2907431c9988b72",
                "subnet-04ad87f71c6e5ab4d",
                "subnet-09d912bb63ef21b9a"
            ],
            "securityGroupIds": [
                "sg-0c1327f6270afbb36"
            ],
            "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
            "vpcId": "vpc-0012b8e1cc0abb17d",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": true,
            "publicAccessCidrs": [
                "0.0.0.0/0"
            ]
        },
        "kubernetesNetworkConfig": {
            "serviceIpv4Cidr": "10.100.0.0/16",
            "ipFamily": "ipv4"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": true
                }
            ]
        },
        "identity": {
            "oidc": {
                "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
DALSJ343KE23J3RN45653DSKJTT647TYD"
```

```
            }
        },
        "status": "DELETING",
        "certificateAuthority": {
            "data": "XXX_CA_DATA_XXX"
        },
        "platformVersion": "eks.16",
        "tags": {
            "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
            "alpha.eksctl.io/cluster-name": "my-eks-cluster",
            "karpenter.sh/discovery": "my-eks-cluster",
            "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
            "auto-delete": "no",
            "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
            "EKS-Cluster-Name": "my-eks-cluster",
            "alpha.eksctl.io/cluster-oidc-enabled": "true",
            "aws:cloudformation:logical-id": "ControlPlane",
            "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
            "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
        },
        "accessConfig": {
            "authenticationMode": "API_AND_CONFIG_MAP"
        }
    }
}
```

Pour plus d'informations, consultez Supprimer un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DeleteClusterla section Référence des AWS CLI commandes.

`delete-fargate-profile`

L'exemple de code suivant montre comment utiliser`delete-fargate-profile`.

AWS CLI

Exemple 1 : créer un profil EKS Fargate pour un sélecteur avec un espace de noms

L'`delete-fargate-profile`exemple suivant crée un profil Fargate EKS pour un sélecteur avec un espace de noms.

```
aws eks delete-fargate-profile \
    --cluster-name my-eks-cluster \
    --fargate-profile-name my-fargate-profile
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/1ac72bb3-3fc6-2631-f1e1-98bff53bed62",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T11:48:39.975000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default",
                "labels": {
                    "foo": "bar"
                }
            }
        ],
        "status": "DELETING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Profil AWS Fargate - Suppression d'une Fargate dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DeleteFargateProfilela section Référence des AWS CLI commandes.

`delete-nodegroup`

L'exemple de code suivant montre comment utiliser`delete-nodegroup`.

AWS CLI

Exemple 1 : supprimer un groupe de nœuds gérés pour un cluster Amazon EKS

L'`delete-nodegroup`exemple suivant supprime un groupe de nœuds gérés pour un cluster Amazon EKS.

```
aws eks delete-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup
```

Sortie :

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/1ec75f5f-0e21-dcc0-b46e-f9c442685cd8",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-08T13:25:15.033000-04:00",
        "modifiedAt": "2024-04-08T13:25:31.252000-04:00",
        "status": "DELETING",
        "capacityType": "SPOT",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 5,
            "desiredSize": 4
        },
        "instanceTypes": [
            "t3.large"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
```

```
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "labels": {
            "my-eks-nodegroup-label-2": "value-2",
            "my-eks-nodegroup-label-1": "value-1"
        },
        "taints": [
            {
                "key": "taint-key-1",
                "value": "taint-value-1",
                "effect": "NO_EXECUTE"
            }
        ],
        "diskSize": 50,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 2
        },
        "tags": {
            "my-eks-nodegroup-key-1": "value-1",
            "my-eks-nodegroup-key-2": "value-2"
        }
    }
}
```

- Pour plus de détails sur l'API, voir DeleteNodegroupla section Référence des AWS CLI commandes.

`delete-pod-identity-association`

L'exemple de code suivant montre comment utiliser`delete-pod-identity-association`.

AWS CLI

Pour supprimer l'association EKS Pod Identity

L'`delete-pod-identity-association`exemple suivant supprime l'association EKS Pod Identity avec l'ID d'association `a-9njjin9gfghecgocd` du cluster EKS nommé`eks-customer`.

```
aws eks delete-pod-identity-association \
    --cluster-name eks-customer \
```

```
    --association-id a-9njjin9gfghecgocd
```

Sortie :

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/s3-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-25T21:10:56.923000-05:00"
    }
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DeletePodIdentityAssociationla section Référence des AWS CLI commandes.

`deregister-cluster`

L'exemple de code suivant montre comment utiliser`deregister-cluster`.

AWS CLI

Pour annuler l'enregistrement d'un cluster connecté afin de le supprimer du plan de contrôle Amazon EKS

L'`deregister-cluster`exemple suivant annule l'enregistrement d'un cluster connecté pour le supprimer du plan de contrôle Amazon EKS.

```
aws eks deregister-cluster \
    --name my-eks-anywhere-cluster
```

Sortie :

```
{
    "cluster": {
        "name": "my-eks-anywhere-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
        "createdAt": "2024-04-12T12:38:37.561000-04:00",
        "status": "DELETING",
        "tags": {},
        "connectorConfig": {
            "activationId": "dfb5ad28-13c3-4e26-8a19-5b2457638c74",
            "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
            "provider": "EKS_ANYWHERE",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

Pour plus d'informations, consultez la section Annulation de l'enregistrement d'un cluster dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DeregisterClusterla section Référence des AWS CLI commandes.

`describe-access-entry`

L'exemple de code suivant montre comment utiliser`describe-access-entry`.

AWS CLI

Pour décrire l'entrée d'accès pour le cluster EKS

L'`describe-access-entry`exemple suivant décrit une entrée d'accès pour le cluster EKS.

```
aws eks describe-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:user/eks-admin-user
```

Sortie :

```
{
    "accessEntry": {
```

```
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:user/eks-admin-user",
        "kubernetesGroups": [],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/user/111122223333/eks-admin-user/0acb1bc6-cb0a-ede6-11ae-a6506e3d36p0",
        "createdAt": "2025-04-14T22:45:48.097000-05:00",
        "modifiedAt": "2025-04-14T22:45:48.097000-05:00",
        "tags": {},
        "username": "arn:aws:iam::111122223333:user/eks-admin-user",
        "type": "STANDARD"
    }
}
```

Pour plus d'informations, consultez la section Accorder aux utilisateurs IAM l'accès à Kubernetes avec des entrées d'accès EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DescribeAccessEntryla section Référence des AWS CLI commandes.

`describe-addon-configuration`

L'exemple de code suivant montre comment utiliser`describe-addon-configuration`.

AWS CLI

Exemple 1 : Options de configuration disponibles lors de la création ou de la mise à jour d'Amazon vpc-cni AddOns

L'`describe-addon-configuration`exemple suivant renvoie tous les schémas de configuration disponibles que vous utilisez lorsqu'un module complémentaire est créé ou mis à jour pour le module complémentaire vpc-cni avec la version correspondante.

```
aws eks describe-addon-configuration \
    --addon-name vpc-cni \
    --addon-version v1.15.1-eksbuild.1
```

Sortie :

```
{
    "addonName": "vpc-cni",
    "addonVersion": "v1.15.1-eksbuild.1",
```

```
    "configurationSchema": "{\"$ref\":\"#/definitions/VpcCni\",\"$schema\":\"http://
json-schema.org/draft-06/schema#\",\"definitions\":{\"Affinity\":{\"type\":
[\"object\",\"null\"]},\"EniConfig\":{\"additionalProperties\":false,\"properties
\":{\"create\":{\"type\":\"boolean\"},\"region\":{\"type\":\"string\"},\"subnets\":
{\"additionalProperties\":{\"additionalProperties\":false,\"properties\":{\"id\":
{\"type\":\"string\"},\"securityGroups\":{\"items\":{\"type\":\"string\"},\"type\":
\"array\"}},\"required\":[\"id\"],\"type\":\"object\"},\"minProperties\":1,\"type
\":\"object\"}},\"required\":[\"create\",\"region\",\"subnets\"],\"type\":\"object
\"},\"Env\":{\"additionalProperties\":false,\"properties\":{\"ADDITIONAL_ENI_TAGS
\":{\"type\":\"string\"},\"ANNOTATE_POD_IP\":{\"format\":\"boolean\",\"type\":
\"string\"},\"AWS_EC2_ENDPOINT\":{\"type\":\"string\"},\"AWS_EXTERNAL_SERVICE_CIDRS
\":{\"type\":\"string\"},\"AWS_MANAGE_ENIS_NON_SCHEDULABLE\":{\"format\":\"boolean
\",\"type\":\"string\"},\"AWS_VPC_CNI_NODE_PORT_SUPPORT\":{\"format\":\"boolean
\",\"type\":\"string\"},\"AWS_VPC_ENI_MTU\":{\"format\":\"integer\",\"type\":
\"string\"},\"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\":{\"format\":\"boolean\",\"type
\":\"string\"},\"AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS\":{\"type\":\"string\"},
\"AWS_VPC_K8S_CNI_EXTERNALSNAT\":{\"format\":\"boolean\",\"type\":\"string\"},
\"AWS_VPC_K8S_CNI_LOGLEVEL\":{\"type\":\"string\"},\"AWS_VPC_K8S_CNI_LOG_FILE\":
{\"type\":\"string\"},\"AWS_VPC_K8S_CNI_RANDOMIZESNAT\":{\"type\":\"string\"},
\"AWS_VPC_K8S_CNI_VETHPREFIX\":{\"type\":\"string\"},\"AWS_VPC_K8S_PLUGIN_LOG_FILE
\":{\"type\":\"string\"},\"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\":{\"type\":\"string
\"},\"CLUSTER_ENDPOINT\":{\"type\":\"string\"},\"DISABLE_INTROSPECTION\":
{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_LEAKED_ENI_CLEANUP\":
{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_METRICS\":{\"format
\":\"boolean\",\"type\":\"string\"},\"DISABLE_NETWORK_RESOURCE_PROVISIONING
\":{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_POD_V6\":{\"format
\":\"boolean\",\"type\":\"string\"},\"ENABLE_BANDWIDTH_PLUGIN\":{\"format\":
\"boolean\",\"type\":\"string\"},\"ENABLE_POD_ENI\":{\"format\":\"boolean\",
\"type\":\"string\"},\"ENABLE_PREFIX_DELEGATION\":{\"format\":\"boolean\",
\"type\":\"string\"},\"ENABLE_V4_EGRESS\":{\"format\":\"boolean\",\"type\":
\"string\"},\"ENABLE_V6_EGRESS\":{\"format\":\"boolean\",\"type\":\"string\"},
\"ENI_CONFIG_ANNOTATION_DEF\":{\"type\":\"string\"},\"ENI_CONFIG_LABEL_DEF\":
{\"type\":\"string\"},\"INTROSPECTION_BIND_ADDRESS\":{\"type\":\"string\"},
\"IP_COOLDOWN_PERIOD\":{\"format\":\"integer\",\"type\":\"string\"},\"MAX_ENI
\":{\"format\":\"integer\",\"type\":\"string\"},\"MINIMUM_IP_TARGET\":{\"format
\":\"integer\",\"type\":\"string\"},\"POD_SECURITY_GROUP_ENFORCING_MODE\":
{\"type\":\"string\"},\"WARM_ENI_TARGET\":{\"format\":\"integer\",\"type\":
\"string\"},\"WARM_IP_TARGET\":{\"format\":\"integer\",\"type\":\"string\"},
\"WARM_PREFIX_TARGET\":{\"format\":\"integer\",\"type\":\"string\"}},\"title
\":\"Env\",\"type\":\"object\"},\"Init\":{\"additionalProperties\":false,
\"properties\":{\"env\":{\"$ref\":\"#/definitions/InitEnv\"}},\"title\":\"Init
\",\"type\":\"object\"},\"InitEnv\":{\"additionalProperties\":false,\"properties
\":{\"DISABLE_TCP_EARLY_DEMUX\":{\"format\":\"boolean\",\"type\":\"string\"},
\"ENABLE_V6_EGRESS\":{\"format\":\"boolean\",\"type\":\"string\"}},\"title\":
```

```
\"InitEnv\",\"type\":\"object\"},\"Limits\":{\"additionalProperties\":false,
\"properties\":{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type\":\"string\"}},
\"title\":\"Limits\",\"type\":\"object\"},\"NodeAgent\":{\"additionalProperties
\":false,\"properties\":{\"enableCloudWatchLogs\":{\"format\":\"boolean\",
\"type\":\"string\"},\"enablePolicyEventLogs\":{\"format\":\"boolean\",\"type\":
\"string\"},\"healthProbeBindAddr\":{\"format\":\"integer\",\"type\":\"string
\"},\"metricsBindAddr\":{\"format\":\"integer\",\"type\":\"string\"}},\"title\":
\"NodeAgent\",\"type\":\"object\"},\"Resources\":{\"additionalProperties\":false,
\"properties\":{\"limits\":{\"$ref\":\"#/definitions/Limits\"},\"requests\":
{\"$ref\":\"#/definitions/Limits\"}},\"title\":\"Resources\",\"type\":\"object
\"},\"Tolerations\":{\"additionalProperties\":false,\"items\":{\"type\":\"object
\"},\"type\":\"array\"},\"VpcCni\":{\"additionalProperties\":false,\"properties
\":{\"affinity\":{\"$ref\":\"#/definitions/Affinity\"},\"enableNetworkPolicy\":
{\"format\":\"boolean\",\"type\":\"string\"},\"enableWindowsIpam\":{\"format\":
\"boolean\",\"type\":\"string\"},\"eniConfig\":{\"$ref\":\"#/definitions/EniConfig
\"},\"env\":{\"$ref\":\"#/definitions/Env\"},\"init\":{\"$ref\":\"#/definitions/Init
\"},\"livenessProbeTimeoutSeconds\":{\"type\":\"integer\"},\"nodeAgent\":{\"$ref\":
\"#/definitions/NodeAgent\"},\"readinessProbeTimeoutSeconds\":{\"type\":\"integer
\"},\"resources\":{\"$ref\":\"#/definitions/Resources\"},\"tolerations\":{\"$ref
\":\"#/definitions/Tolerations\"}},\"title\":\"VpcCni\",\"type\":\"object\"}},
\"description\":\"vpc-cni\"}"
}
```


Exemple 2 : Options de configuration disponibles lors de la création ou de la mise à jour des coredns Amazon AddOns

L'`describe-addon-configuration`exemple suivant renvoie tous les schémas de configuration disponibles que vous utilisez lorsqu'un module complémentaire est créé ou mis à jour pour le module complémentaire Coredns avec la version correspondante.

```
aws eks describe-addon-configuration \
    --addon-name coredns \
    --addon-version v1.8.7-eksbuild.4
```

Sortie :


```
{
    "addonName": "coredns",
    "addonVersion": "v1.8.7-eksbuild.4",
    "configurationSchema": "{\"$ref\":\"#/definitions/Coredns\",\"$schema
\":\"http://json-schema.org/draft-06/schema#\",\"definitions\":{\"Coredns\":
{\"additionalProperties\":false,\"properties\":{\"computeType\":{\"type\":
```

```
\"string\"},\"corefile\":{\"description\":\"Entire corefile contents to use with
 installation\",\"type\":\"string\"},\"nodeSelector\":{\"additionalProperties\":
{\"type\":\"string\"},\"type\":\"object\"},\"replicaCount\":{\"type\":\"integer
\"},\"resources\":{\"$ref\":\"#/definitions/Resources\"}},\"title\":\"Coredns\",
\"type\":\"object\"},\"Limits\":{\"additionalProperties\":false,\"properties\":
{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type\":\"string\"}},\"title\":\"Limits
\",\"type\":\"object\"},\"Resources\":{\"additionalProperties\":false,\"properties
\":{\"limits\":{\"$ref\":\"#/definitions/Limits\"},\"requests\":{\"$ref\":\"#/
definitions/Limits\"}},\"title\":\"Resources\",\"type\":\"object\"}}}"
}
```

Pour plus d'informations, consultez Création ou mise à jour d'un fichier kubeconfig pour un cluster Amazon EKS dans Amazon EKS.

- Pour plus de détails sur l'API, voir DescribeAddonConfigurationla section Référence des AWS CLI commandes.

`describe-addon-versions`

L'exemple de code suivant montre comment utiliser`describe-addon-versions`.

AWS CLI

Exemple 1 : Répertorier tous les addons disponibles pour EKS Cluster

L'`describe-addon-versions`exemple suivant répertorie tous les AWS addons disponibles.

```
aws eks describe-addon-versions  \
    --query 'sort_by(addons  &owner)[].{publisher: publisher, owner: owner,
 addonName: addonName, type: type}' \
    --output table
```

Sortie :

```
-----------------------------------------------------------------------------------------------
|                                                DescribeAddonVersions
                                |
+---------------------------------------------+------------------
+-----------------------+-------------------------+
|                  addonName                  |      owner       |       publisher
     |          type           |
+---------------------------------------------+------------------
+-----------------------+-------------------------+
```

```
|  vpc-cni                                |  aws             |  eks
     |  networking             |
|  snapshot-controller                    |  aws             |  eks
     |  storage                |
|  kube-proxy                             |  aws             |  eks
     |  networking             |
|  eks-pod-identity-agent                 |  aws             |  eks
     |  security               |
|  coredns                                |  aws             |  eks
     |  networking             |
|  aws-mountpoint-s3-csi-driver           |  aws             |  s3
     |  storage                |
|  aws-guardduty-agent                    |  aws             |  eks
     |  security               |
|  aws-efs-csi-driver                     |  aws             |  eks
     |  storage                |
|  aws-ebs-csi-driver                     |  aws             |  eks
     |  storage                |
|  amazon-cloudwatch-observability        |  aws             |  eks
     |  observability          |
|  adot                                   |  aws             |  eks
     |  observability          |
|  upwind-security_upwind-operator        |  aws-marketplace |  Upwind Security
     |  security               |
|  upbound_universal-crossplane           |  aws-marketplace |  upbound
     |  infra-management       |
|  tetrate-io_istio-distro                |  aws-marketplace |  tetrate-io
     |  policy-management      |
|  teleport_teleport                      |  aws-marketplace |  teleport
     |  policy-management      |
|  stormforge_optimize-live               |  aws-marketplace |  StormForge
     |  cost-management        |
|  splunk_splunk-otel-collector-chart     |  aws-marketplace |  Splunk
     |  monitoring             |
|  solo-io_istio-distro                   |  aws-marketplace |  Solo.io
     |  service-mesh           |
|  rafay-systems_rafay-operator           |  aws-marketplace |  rafay-systems
     |  kubernetes-management  |
|  new-relic_kubernetes-operator          |  aws-marketplace |  New Relic
     |  observability          |
|  netapp_trident-operator                |  aws-marketplace |  NetApp Inc.
     |  storage                |
|  leaksignal_leakagent                   |  aws-marketplace |  leaksignal
     |  monitoring             |
```

```
|  kubecost_kubecost                              |  aws-marketplace |  kubecost
     |  cost-management        |
|  kong_konnect-ri                                |  aws-marketplace |  kong
     |  ingress-service-type   |
|  kasten_k10                                     |  aws-marketplace |  Kasten by Veeam
     |  data-protection        |
|  haproxy-technologies_kubernetes-ingress-ee |  aws-marketplace |  HAProxy
 Technologies |  ingress-controller     |
|  groundcover_agent                              |  aws-marketplace |  groundcover
     |  monitoring             |
|  grafana-labs_kubernetes-monitoring             |  aws-marketplace |  Grafana Labs
     |  monitoring             |
|  factorhouse_kpow                               |  aws-marketplace |  factorhouse
     |  monitoring             |
|  dynatrace_dynatrace-operator                   |  aws-marketplace |  dynatrace
     |  monitoring             |
|  datree_engine-pro                              |  aws-marketplace |  datree
     |  policy-management      |
|  datadog_operator                               |  aws-marketplace |  Datadog
     |  monitoring             |
|  cribl_cribledge                                |  aws-marketplace |  Cribl
     |  observability          |
|  calyptia_fluent-bit                            |  aws-marketplace |  Calyptia Inc
     |  observability          |
|  accuknox_kubearmor                             |  aws-marketplace |  AccuKnox
     |  security               |
+-------------------------------------------------+------------------
+----------------------+-------------------------+
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : Répertorier tous les addons disponibles pour la version spécifiée de Kubernetes prise en charge pour EKS

L'`describe-addon-versions`exemple suivant répertorie tous les addons disponibles pour la version spécifiée de Kubernetes prise en charge pour EKS.

```
aws eks describe-addon-versions  \
    --kubernetes-version=1.26 \
    --query 'sort_by(addons  &owner)[].{publisher: publisher, owner: owner,
 addonName: addonName, type: type}' \
    --output table
```

Sortie :

```
-----------------------------------------------------------------------------------------------------
|                                                DescribeAddonVersions
                                |
+---------------------------------------------+------------------
+-----------------------+------------------------+
|                  addonName                  |      owner       |       publisher
     |          type           |
+---------------------------------------------+------------------
+-----------------------+------------------------+
|  vpc-cni                                    |  aws             |  eks
     |  networking             |
|  snapshot-controller                        |  aws             |  eks
     |  storage                |
|  kube-proxy                                 |  aws             |  eks
     |  networking             |
|  eks-pod-identity-agent                     |  aws             |  eks
     |  security               |
|  coredns                                    |  aws             |  eks
     |  networking             |
|  aws-mountpoint-s3-csi-driver               |  aws             |  s3
     |  storage                |
|  aws-guardduty-agent                        |  aws             |  eks
     |  security               |
|  aws-efs-csi-driver                         |  aws             |  eks
     |  storage                |
|  aws-ebs-csi-driver                         |  aws             |  eks
     |  storage                |
|  amazon-cloudwatch-observability            |  aws             |  eks
     |  observability          |
|  adot                                       |  aws             |  eks
     |  observability          |
|  upwind-security_upwind-operator            |  aws-marketplace |  Upwind Security
     |  security               |
|  tetrate-io_istio-distro                    |  aws-marketplace |  tetrate-io
     |  policy-management      |
|  stormforge_optimize-live                   |  aws-marketplace |  StormForge
     |  cost-management        |
|  splunk_splunk-otel-collector-chart         |  aws-marketplace |  Splunk
     |  monitoring             |
|  solo-io_istio-distro                       |  aws-marketplace |  Solo.io
     |  service-mesh           |
```

```
|  rafay-systems_rafay-operator                |  aws-marketplace |  rafay-systems
     |  kubernetes-management  |
|  new-relic_kubernetes-operator               |  aws-marketplace |  New Relic
     |  observability          |
|  netapp_trident-operator                     |  aws-marketplace |  NetApp Inc.
     |  storage                |
|  leaksignal_leakagent                        |  aws-marketplace |  leaksignal
     |  monitoring             |
|  kubecost_kubecost                           |  aws-marketplace |  kubecost
     |  cost-management        |
|  kong_konnect-ri                             |  aws-marketplace |  kong
     |  ingress-service-type   |
|  haproxy-technologies_kubernetes-ingress-ee |  aws-marketplace |  HAProxy
 Technologies |  ingress-controller     |
|  groundcover_agent                           |  aws-marketplace |  groundcover
     |  monitoring             |
|  grafana-labs_kubernetes-monitoring          |  aws-marketplace |  Grafana Labs
     |  monitoring             |
|  dynatrace_dynatrace-operator                |  aws-marketplace |  dynatrace
     |  monitoring             |
|  datadog_operator                            |  aws-marketplace |  Datadog
     |  monitoring             |
|  cribl_cribledge                             |  aws-marketplace |  Cribl
     |  observability          |
|  calyptia_fluent-bit                         |  aws-marketplace |  Calyptia Inc
     |  observability          |
|  accuknox_kubearmor                          |  aws-marketplace |  AccuKnox
     |  security               |
+----------------------------------------------+------------------
+----------------------+-------------------------+
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : Répertorier toutes les versions d'addons vpc-cni disponibles pour la version Kubernetes spécifiée prise en charge par EKS

L'`describe-addon-versions`exemple suivant répertorie toutes les versions d'addons vpc-cni disponibles pour la version Kubernetes spécifiée prise en charge par EKS.

```
aws eks describe-addon-versions \
    --kubernetes-version=1.26 \
    --addon-name=vpc-cni \
```

```
    --query='addons[].addonVersions[].addonVersion'
```

Sortie :

```
[
    "v1.18.0-eksbuild.1",
    "v1.17.1-eksbuild.1",
    "v1.16.4-eksbuild.2",
    "v1.16.3-eksbuild.2",
    "v1.16.2-eksbuild.1",
    "v1.16.0-eksbuild.1",
    "v1.15.5-eksbuild.1",
    "v1.15.4-eksbuild.1",
    "v1.15.3-eksbuild.1",
    "v1.15.1-eksbuild.1",
    "v1.15.0-eksbuild.2",
    "v1.14.1-eksbuild.1",
    "v1.14.0-eksbuild.3",
    "v1.13.4-eksbuild.1",
    "v1.13.3-eksbuild.1",
    "v1.13.2-eksbuild.1",
    "v1.13.0-eksbuild.1",
    "v1.12.6-eksbuild.2",
    "v1.12.6-eksbuild.1",
    "v1.12.5-eksbuild.2",
    "v1.12.0-eksbuild.2"
]
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Création d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DescribeAddonVersionsla section Référence des AWS CLI commandes.

`describe-addon`

L'exemple de code suivant montre comment utiliser`describe-addon`.

AWS CLI

Décrivez l'exécution active de l'addon EKS dans votre cluster Amazon EKS

L'`describe-addon`exemple suivant exécute activement l'addon EKS dans votre cluster Amazon EKS.

```
aws eks describe-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni
```

Sortie :

```
{
    "addon": {
        "addonName": "vpc-cni",
        "clusterName": "my-eks-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-
cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f",
        "createdAt": "2024-03-14T13:18:45.417000-04:00",
        "modifiedAt": "2024-03-14T13:18:49.557000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-
cluster-addon-vpc-cni-Role1-YfakrqOC1UTm",
        "tags": {
            "eks-addon-key-3": "value-3",
            "eks-addon-key-4": "value-4"
        },
        "configurationValues": "resources:\n    limits:\n        cpu: '100m'\nenv:\n
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
    }
}
```

- Pour plus de détails sur l'API, voir DescribeAddonla section Référence des AWS CLI commandes.

`describe-cluster`

L'exemple de code suivant montre comment utiliser`describe-cluster`.

AWS CLI

Décrivez l'exécution active de l'addon EKS dans votre cluster Amazon EKS

L'`describe-cluster`exemple suivant exécute activement l'addon EKS dans votre cluster Amazon EKS.

```
aws eks describe-cluster \
    --name my-eks-cluster
```

Sortie :

```
{
    "cluster": {
        "name": "my-eks-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
        "createdAt": "2024-03-14T11:31:44.348000-04:00",
        "version": "1.26",
        "endpoint": "https://JSA79429HJDASKJDJ8223829MNDNASW.yl4.us-
east-2.eks.amazonaws.com",
        "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0fb75d2d8401716e7",
                "subnet-02184492f67a3d0f9",
                "subnet-04098063527aab776",
                "subnet-0e2907431c9988b72",
                "subnet-04ad87f71c6e5ab4d",
                "subnet-09d912bb63ef21b9a"
            ],
            "securityGroupIds": [
                "sg-0c1327f6270afbb36"
            ],
            "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
            "vpcId": "vpc-0012b8e1cc0abb17d",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": true,
            "publicAccessCidrs": [
                "22.19.18.2/32"
            ]
        },
        "kubernetesNetworkConfig": {
```

```
            "serviceIpv4Cidr": "10.100.0.0/16",
            "ipFamily": "ipv4"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": true
                }
            ]
        },
        "identity": {
            "oidc": {
                "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
JSA79429HJDASKJDJ8223829MNDNASW"
            }
        },
        "status": "ACTIVE",
        "certificateAuthority": {
            "data": "CA_DATA_STRING..."
        },
        "platformVersion": "eks.14",
        "tags": {
            "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
            "alpha.eksctl.io/cluster-name": "my-eks-cluster",
            "karpenter.sh/discovery": "my-eks-cluster",
            "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
            "auto-delete": "no",
            "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
            "EKS-Cluster-Name": "my-eks-cluster",
            "alpha.eksctl.io/cluster-oidc-enabled": "true",
            "aws:cloudformation:logical-id": "ControlPlane",
            "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
            "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
        },
```

```
        "health": {
            "issues": []
        },
        "accessConfig": {
            "authenticationMode": "API_AND_CONFIG_MAP"
        }
    }
}
```

- Pour plus de détails sur l'API, voir DescribeClusterla section Référence des AWS CLI commandes.

`describe-fargate-profile`

L'exemple de code suivant montre comment utiliser`describe-fargate-profile`.

AWS CLI

Décrire un profil Fargate

L'`describe-fargate-profile`exemple suivant décrit un profil Fargate.

```
aws eks describe-fargate-profile \
    --cluster-name my-eks-cluster \
    --fargate-profile-name my-fargate-profile
```

Sortie :

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/96c766ce-43d2-f9c9-954c-647334391198",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-04-11T10:42:52.486000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-
cluster-farga-FargatePodExecutionRole-1htfAaJdJUEO",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
```

```
        ],
        "selectors": [
            {
                "namespace": "prod*",
                "labels": {
                    "labelname*?": "*value1"
                }
            },
            {
                "namespace": "*dev*",
                "labels": {
                    "labelname*?": "*value*"
                }
            }
        ],
        "status": "ACTIVE",
        "tags": {
            "eks-fargate-profile-key-2": "value-2",
            "eks-fargate-profile-key-1": "value-1"
        }
    }
}
```

- Pour plus de détails sur l'API, voir DescribeFargateProfilela section Référence des AWS CLI commandes.

`describe-identity-provider-config`

L'exemple de code suivant montre comment utiliser`describe-identity-provider-config`.

AWS CLI

Décrivez une configuration de fournisseur d'identité associée à votre cluster Amazon EKS

L'`describe-identity-provider-config`exemple suivant décrit une configuration de fournisseur d'identité associée à votre cluster Amazon EKS.

```
aws eks describe-identity-provider-config \
    --cluster-name my-eks-cluster \
    --identity-provider-config type=oidc,name=my-identity-provider
```

Sortie :

```
{
    "identityProviderConfig": {
        "oidc": {
            "identityProviderConfigName": "my-identity-provider",
            "identityProviderConfigArn": "arn:aws:eks:us-
east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-
provider/8ac76722-78e4-cec1-ed76-d49eea058622",
            "clusterName": "my-eks-cluster",
            "issuerUrl": "https://oidc.eks.us-east-2.amazonaws.com/
id/38D6A4619A0A69E342B113ED7F1A7652",
            "clientId": "kubernetes",
            "usernameClaim": "email",
            "usernamePrefix": "my-username-prefix",
            "groupsClaim": "my-claim",
            "groupsPrefix": "my-groups-prefix",
            "requiredClaims": {
                "Claim1": "value1",
                "Claim2": "value2"
            },
            "tags": {
                "env": "dev"
            },
            "status": "ACTIVE"
        }
    }
}
```

Pour plus d'informations, consultez Authentifier les utilisateurs de votre cluster auprès d'un fournisseur d'identité OpenID Connect dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DescribeIdentityProviderConfigla section Référence des AWS CLI commandes.

`describe-insight`

L'exemple de code suivant montre comment utiliser`describe-insight`.

AWS CLI

Pour obtenir les détails d'un aperçu d'un cluster EKS à l'aide de son ID

L'`describe-insight`exemple suivant renvoie les détails relatifs à l'aperçu spécifié à l'aide du nom du cluster et de l'identifiant d'aperçu.

```
aws eks describe-insight \
    --cluster-name eks-customer \
    --id 38ea7a64-a14f-4e0e-95c7-8dbcab3c3623
```

Sortie :

```
{
    "insight": {
        "id": "38ea7a64-a14f-4e0e-95c7-8dbcab3c3623",
        "name": "Kubelet version skew",
        "category": "UPGRADE_READINESS",
        "kubernetesVersion": "1.33",
        "lastRefreshTime": "2025-05-24T11:22:50-05:00",
        "lastTransitionTime": "2025-05-24T11:22:50-05:00",
        "description": "Checks for kubelet versions of worker nodes in the cluster
 to see if upgrade would cause noncompliance with supported Kubernetes kubelet
 version skew policy.",
        "insightStatus": {
            "status": "PASSING",
            "reason": "Node kubelet versions match the cluster control plane
 version."
        },
        "recommendation": "Upgrade your worker nodes to match the Kubernetes version
 of your cluster control plane.",
        "additionalInfo": {
            "Kubelet version skew policy": "https://kubernetes.io/releases/version-
skew-policy/#kubelet",
            "Updating a managed node group": "https://docs.aws.amazon.com/eks/
latest/userguide/update-managed-node-group.html"
        },
        "resources": [],
        "categorySpecificSummary": {
            "deprecationDetails": []
        }
    }
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les clusters dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DescribeInsightla section Référence des AWS CLI commandes.

`describe-nodegroup`

L'exemple de code suivant montre comment utiliser`describe-nodegroup`.

AWS CLI

Décrire un groupe de nœuds géré pour un cluster Amazon EKS

L'`describe-nodegroup`exemple suivant décrit un groupe de nœuds géré pour un cluster Amazon EKS.

```
aws eks describe-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup
```

Sortie :

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/a8c75f2f-df78-a72f-4063-4b69af3de5b1",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-08T11:42:10.555000-04:00",
        "modifiedAt": "2024-04-08T11:44:12.402000-04:00",
        "status": "ACTIVE",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 3,
            "desiredSize": 1
        },
        "instanceTypes": [
            "t3.medium"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
```

```
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "labels": {},
        "resources": {
            "autoScalingGroups": [
                {
                    "name": "eks-my-eks-nodegroup-a8c75f2f-df78-
a72f-4063-4b69af3de5b1"
                }
            ]
        },
        "diskSize": 20,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "tags": {}
    }
}
```

- Pour plus de détails sur l'API, voir DescribeNodegroupla section Référence des AWS CLI commandes.

`describe-pod-identity-association`

L'exemple de code suivant montre comment utiliser`describe-pod-identity-association`.

AWS CLI

Pour fournir des informations sur l'association Pod Identity

L'`describe-pod-identity-association`exemple suivant décrit une association Pod Identity dans le cluster EKS.

```
aws eks describe-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd
```

Sortie :

```
{
```

```
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-24T19:52:14.135000-05:00"
    }
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DescribePodIdentityAssociationla section Référence des AWS CLI commandes.

`describe-update`

L'exemple de code suivant montre comment utiliser`describe-update`.

AWS CLI

Exemple 1 : pour décrire une mise à jour pour un cluster

L'`describe-update`exemple suivant décrit une mise à jour pour un cluster nommé.

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161
```

Sortie :

```
{
    "update": {
        "id": "10bddb13-a71b-425a-b0a6-71cd03e59161",
```

```
        "status": "Successful",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "false"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            }
        ],
        "createdAt": "2024-03-14T10:01:26.297000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une version Kubernetes d'un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : pour décrire une mise à jour pour un cluster

L'`describe-update`exemple suivant décrit une mise à jour pour un cluster nommé.

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id e4994991-4c0f-475a-a040-427e6da52966
```

Sortie :

```
{
    "update": {
        "id": "e4994991-4c0f-475a-a040-427e6da52966",
        "status": "Successful",
        "type": "AssociateEncryptionConfig",
        "params": [
            {
                "type": "EncryptionConfig",
                "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
            }
        ],
        "createdAt": "2024-03-14T11:01:26.297000-04:00",
```

```
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une version Kubernetes d'un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : pour décrire une mise à jour pour un cluster

L'`describe-update`exemple suivant décrit une mise à jour pour un cluster nommé.

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

Sortie :

```
{
    "update": {
        "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
        "status": "Successful",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.29"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.1"
            }
        ],
        "createdAt": "2024-03-14T12:05:26.297000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une version Kubernetes d'un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DescribeUpdatela section Référence des AWS CLI commandes.

`disassociate-access-policy`

L'exemple de code suivant montre comment utiliser`disassociate-access-policy`.

AWS CLI

Pour dissocier la politique d'accès d'une entrée d'accès

Ce qui suit `disassociate-access-policy` supprime la politique d'accès associée à l'entrée d'accès.

```
aws eks disassociate-access-policy \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Associer les politiques d'accès aux entrées d'accès dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DisassociateAccessPolicyla section Référence des AWS CLI commandes.

`disassociate-identity-provider-config`

L'exemple de code suivant montre comment utiliser`disassociate-identity-provider-config`.

AWS CLI

Dissocier le fournisseur d'identité de votre cluster Amazon EKS

L'`disassociate-identity-provider-config`exemple suivant dissocie un fournisseur d'identité de votre cluster Amazon EKS.

```
aws eks disassociate-identity-provider-config \
    --cluster-name my-eks-cluster \
    --identity-provider-config 'type=oidc,name=my-identity-provider'
```

Sortie :

```
{
    "update": {
        "id": "5f78d14e-c57b-4857-a3e4-cf664ae20949",
        "status": "InProgress",
        "type": "DisassociateIdentityProviderConfig",
        "params": [
            {
                "type": "IdentityProviderConfig",
                "value": "[]"
            }
        ],
        "createdAt": "2024-04-11T13:53:43.314000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez Authentifier les utilisateurs de votre cluster auprès d'un fournisseur d'identité OpenID Connect - Dissocier un fournisseur d'identité OIDC de votre cluster dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir DisassociateIdentityProviderConfigla section Référence des AWS CLI commandes.

`get-token`

L'exemple de code suivant montre comment utiliser`get-token`.

AWS CLI

Exemple 1 : obtenir un jeton d'authentification pour un cluster Amazon EKS nommé `my-eks-cluster`

L'`get-token`exemple suivant obtient un jeton d'authentification pour un cluster Amazon EKS nommé my-eks-cluster.

```
aws eks get-token \
    --cluster-name my-eks-cluster
```

Sortie :

```
{
```

```
    "kind": "ExecCredential",
    "apiVersion": "client.authentication.k8s.io/v1beta1",
    "spec": {},
    "status": {
        "expirationTimestamp": "2024-04-11T20:59:56Z",
        "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
    }
}
```

Exemple 2 : obtient un jeton d'authentification pour un cluster Amazon EKS nommé `my-eks-cluster` en utilisant ce roLearn pour les informations d'identification lors de la signature du jeton

L'`get-token`exemple suivant obtient un jeton d'authentification pour un cluster Amazon EKS nommé my-eks-cluster en utilisant ce ROLearn pour les informations d'identification lors de la signature du jeton.

```
aws eks get-token \
    --cluster-name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-ServiceRole-j1k7AfTIQtnM
```

Sortie :

```
{
    "kind": "ExecCredential",
    "apiVersion": "client.authentication.k8s.io/v1beta1",
    "spec": {},
    "status": {
        "expirationTimestamp": "2024-04-11T21:05:26Z",
        "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
    }
}
```

- Pour plus de détails sur l'API, voir GetTokenla section Référence des AWS CLI commandes.

`list-access-entries`

L'exemple de code suivant montre comment utiliser`list-access-entries`.

AWS CLI

Pour répertorier les entrées d'accès pour un cluster EKS

Ce qui suit `list-access-entries` renvoie la liste des entrées d'accès associées au cluster EKS`eks-customer`.

```
aws eks list-access-entries \
    --cluster-name eks-customer
```

Sortie :

```
{
    "accessEntries": [
        "arn:aws:iam::111122223333:role/Admin",
        "arn:aws:iam::111122223333:role/admin-test-ip",
        "arn:aws:iam::111122223333:role/assume-worker-node-role",
        "arn:aws:iam::111122223333:user/eks-admin-user"
    ]
}
```

Pour plus d'informations, consultez la section Accorder aux utilisateurs IAM l'accès à Kubernetes avec des entrées d'accès EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir ListAccessEntriesla section Référence des AWS CLI commandes.

`list-access-policies`

L'exemple de code suivant montre comment utiliser`list-access-policies`.

AWS CLI

Pour répertorier toutes les politiques d'accès disponibles

Cet `list-access-policies` exemple renvoie la liste de toutes les politiques d'accès disponibles.

```
aws eks list-access-policies
```

Sortie :

```
{
    "accessPolicies": [
        {
            "name": "AmazonEKSAdminPolicy",
```

```
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy"
        },
        {
            "name": "AmazonEKSAdminViewPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminViewPolicy"
        },
        {
            "name": "AmazonEKSAutoNodePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAutoNodePolicy"
        },
        {
            "name": "AmazonEKSBlockStorageClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSBlockStorageClusterPolicy"
        },
        {
            "name": "AmazonEKSBlockStoragePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSBlockStoragePolicy"
        },
        {
            "name": "AmazonEKSClusterAdminPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSClusterAdminPolicy"
        },
        {
            "name": "AmazonEKSComputeClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSComputeClusterPolicy"
        },
        {
            "name": "AmazonEKSComputePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSComputePolicy"
        },
        {
            "name": "AmazonEKSEditPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy"
        },
        {
            "name": "AmazonEKSHybridPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSHybridPolicy"
        },
        {
            "name": "AmazonEKSLoadBalancingClusterPolicy",
```

```
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSLoadBalancingClusterPolicy"
        },
        {
            "name": "AmazonEKSLoadBalancingPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSLoadBalancingPolicy"
        },
        {
            "name": "AmazonEKSNetworkingClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSNetworkingClusterPolicy"
        },
        {
            "name": "AmazonEKSNetworkingPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSNetworkingPolicy"
        },
        {
            "name": "AmazonEKSViewPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy"
        },
        {
            "name": "AmazonEMRJobPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEMRJobPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodClusterPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodControllerPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodControllerPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodSystemNamespacePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodSystemNamespacePolicy"
        }
    ]
}
```

Pour plus d'informations, consultez Associer les politiques d'accès aux entrées d'accès dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir ListAccessPolicies la section Référence des AWS CLI commandes.

`list-addons`

L'exemple de code suivant montre comment utiliser `list-addons`.

AWS CLI

Répertoriez tous les modules complémentaires installés dans votre cluster Amazon EKS nommé `my-eks-cluster`

L'`list-addons` exemple suivant répertorie tous les modules complémentaires installés dans votre cluster Amazon EKS nommé my-eks-cluster.

```
aws eks list-addons \
    --cluster-name my-eks-cluster
```

Sortie :

```
{
    "addons": [
        "kube-proxy",
        "vpc-cni"
    ]
}
```

- Pour plus de détails sur l'API, voir ListAddons la section Référence des AWS CLI commandes.

`list-associated-access-policies`

L'exemple de code suivant montre comment utiliser `list-associated-access-policies`.

AWS CLI

Pour répertorier les politiques d'accès associées à une entrée d'accès

L'`list-associated-access-policies` exemple suivant renvoie la liste des politiques d'accès associées à une entrée d'accès dans le cluster EKS.

```
aws eks list-associated-access-policies \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin
```

Sortie :

```
{
    "associatedAccessPolicies": [
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSAdminPolicy",
            "accessScope": {
                "type": "cluster",
                "namespaces": []
            },
            "associatedAt": "2025-05-24T17:26:22.935000-05:00",
            "modifiedAt": "2025-05-24T17:26:22.935000-05:00"
        }
    ],
    "clusterName": "eks-customer",
    "principalArn": "arn:aws:iam::111122223333:role/Admin"
}
```

Pour plus d'informations, consultez la section Accorder aux utilisateurs IAM l'accès à Kubernetes avec des entrées d'accès EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir ListAssociatedAccessPolicieslа section Référence des AWS CLI commandes.

`list-clusters`

L'exemple de code suivant montre comment utiliser`list-clusters`.

AWS CLI

Pour répertorier tous les modules complémentaires installés dans votre cluster Amazon EKS nommé `my-eks-cluster`

L'`list-clusters`exemple suivant répertorie tous les modules complémentaires installés dans votre cluster Amazon EKS nommé my-eks-cluster.

```
aws eks list-clusters
```

Sortie :

```
{
    "clusters": [
        "prod",
        "qa",
        "stage",
        "my-eks-cluster"
    ]
}
```

- Pour plus de détails sur l'API, voir ListClustersla section Référence des AWS CLI commandes.

`list-fargate-profiles`

L'exemple de code suivant montre comment utiliser`list-fargate-profiles`.

AWS CLI

Pour répertorier tous les profils fargate de votre cluster Amazon EKS nommé `` my-eks-cluster

L'`list-fargate-profiles`exemple suivant répertorie tous les profils fargate de votre cluster Amazon EKS nommé. my-eks-cluster

```
aws eks list-fargate-profiles \
    --cluster-name my-eks-cluster
```

Sortie :

```
{
    "fargateProfileNames": [
        "my-fargate-profile"
    ]
}
```

- Pour plus de détails sur l'API, voir ListFargateProfilesla section Référence des AWS CLI commandes.

`list-identity-provider-configs`

L'exemple de code suivant montre comment utiliser`list-identity-provider-configs`.

AWS CLI

Répertorier les fournisseurs d'identité associés à un cluster Amazon EKS

L'`list-identity-provider-configs`exemple suivant répertorie le fournisseur d'identité associé à un cluster Amazon EKS.

```
aws eks list-identity-provider-configs \
    --cluster-name my-eks-cluster
```

Sortie :

```
{
    "identityProviderConfigs": [
        {
            "type": "oidc",
            "name": "my-identity-provider"
        }
    ]
}
```

Pour plus d'informations, consultez Authentifier les utilisateurs de votre cluster auprès d'un fournisseur d'identité OpenID Connect dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir ListIdentityProviderConfigsla section Référence des AWS CLI commandes.

`list-insights`

L'exemple de code suivant montre comment utiliser`list-insights`.

AWS CLI

Pour répertorier toutes les informations relatives au cluster spécifié

L'`list-insights`exemple suivant renvoie la liste de toutes les informations vérifiées par rapport au cluster spécifié.

```
aws eks list-insights \
--cluster-name eks-customer
```

Sortie :


```
{
    "insights": [
        {
            "id": "38ea7a64-a14f-4e0e-95c7-8dbcab3c3616",
            "name": "Kubelet version skew",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:50-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks for kubelet versions of worker nodes in the
 cluster to see if upgrade would cause noncompliance with supported Kubernetes
 kubelet version skew policy.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "Node kubelet versions match the cluster control plane
 version."
            }
        },
        {
            "id": "9cd91472-f99c-45a9-b7d7-54d4900dee23",
            "name": "EKS add-on version compatibility",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:59-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks version of installed EKS add-ons to ensure they
 are compatible with the next version of Kubernetes. ",
            "insightStatus": {
                "status": "PASSING",
                "reason": "All installed EKS add-on versions are compatible with
 next Kubernetes version."
            }
        },
        {
            "id": "0deb269d-b1e1-458c-a2b4-7a57f940c875",
            "name": "Cluster health issues",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
```

```
            "lastRefreshTime": "2025-05-24T11:22:59-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks for any cluster health issues that prevent
 successful upgrade to the next Kubernetes version on EKS.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "No cluster health issues detected."
            }
        },
        {
            "id": "214fa274-344f-420b-812a-5049ce72c9ww",
            "name": "kube-proxy version skew",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:50-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks version of kube-proxy in cluster to see if
 upgrade would cause noncompliance with supported Kubernetes kube-proxy version skew
 policy.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "kube-proxy versions match the cluster control plane
 version."
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les clusters dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, voir ListInsightsla section Référence des AWS CLI commandes.

`list-nodegroups`

L'exemple de code suivant montre comment utiliser`list-nodegroups`.

AWS CLI

Répertorier tous les groupes de nœuds d'un cluster Amazon EKS

L'`list-nodegroups`exemple suivant répertorie tous les groupes de nœuds d'un cluster Amazon EKS.

```
aws eks list-nodegroups \
    --cluster-name my-eks-cluster
```

Sortie :

```
{
    "nodegroups": [
        "my-eks-managed-node-group",
        "my-eks-nodegroup"
    ]
}
```

- Pour plus de détails sur l'API, voir ListNodegroupsla section Référence des AWS CLI commandes.

`list-pod-identity-associations`

L'exemple de code suivant montre comment utiliser`list-pod-identity-associations`.

AWS CLI

Exemple 1 : pour répertorier les associations Pod Identity dans un cluster EKS

Ce qui suit `list-pod-identity-associations` renvoie la liste des associations Pod Identity associées au cluster EKS nommé `eks-customer` dans tous les espaces de noms et comptes de service.

```
aws eks list-pod-identity-associations \
    --cluster-name eks-customer
```

Sortie :

```
{
    "associations": [
        {
            "clusterName": "eks-customer",
            "namespace": "default",
            "serviceAccount": "default",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
```

```
            "associationId": "a-9njjin9gfghecgocd"
        },
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "eks-customer",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-dvtacahdvjn01ffbc",
            "associationId": "a-dvtacahdvjn01ffbc"
        },
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "coredns",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-yrpsdroc4ei7k6xps",
            "associationId": "a-yrpsdroc4ei7k6xps"
        }
    ]
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : pour répertorier les associations Pod Identity dans un cluster EKS en fonction de l'espace de noms et du compte de service

Ce qui suit `list-pod-identity-associations` renvoie la liste des associations Pod Identity dans le cluster EKS en fonction de l'espace de noms et du compte de service.

```
aws eks list-pod-identity-associations \
    --cluster-name eks-customer \
    --namespace kube-system \
    --service-account eks-customer
```

Sortie :

```
{
    "associations": [
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
```

```
            "serviceAccount": "eks-customer",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-dvtacahdvjn01ffbc",
            "associationId": "a-dvtacahdvjn01ffbc"
        }
    ]
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous ListPodIdentityAssociationsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Exemple 1 : pour répertorier toutes les balises d'un ARN de cluster Amazon EKS

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'un ARN de cluster Amazon EKS.

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

Sortie :

```
{
    "tags": {
        "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
        "alpha.eksctl.io/cluster-name": "my-eks-cluster",
        "karpenter.sh/discovery": "my-eks-cluster",
        "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
        "auto-delete": "no",
        "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
        "EKS-Cluster-Name": "my-eks-cluster",
```

```
        "alpha.eksctl.io/cluster-oidc-enabled": "true",
        "aws:cloudformation:logical-id": "ControlPlane",
        "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
        "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
    }
}
```

Exemple 2 : pour répertorier toutes les balises d'un groupe de nœuds Amazon EKS (ARN)

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées à l'ARN d'un groupe de nœuds Amazon EKS.

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c
```

Sortie :

```
{
    "tags": {
        "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-nodegroup-my-eks-
managed-node-group",
        "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-nodegroup-my-eks-managed-node-group/
eaa20310-e219-11ee-b851-0ab9ad8228ff",
        "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
        "EKS-Cluster-Name": "my-eks-cluster",
        "alpha.eksctl.io/nodegroup-type": "managed",
        "NodeGroup Name 1": "my-eks-managed-node-group",
        "k8s.io/cluster-autoscaler/enabled": "true",
        "nodegroup-role": "worker",
        "alpha.eksctl.io/cluster-name": "my-eks-cluster",
        "alpha.eksctl.io/nodegroup-name": "my-eks-managed-node-group",
        "karpenter.sh/discovery": "my-eks-cluster",
        "NodeGroup Name 2": "AmazonLinux-Linux-Managed-NG-v1-26-v1",
        "auto-delete": "no",
        "k8s.io/cluster-autoscaler/my-eks-cluster": "owned",
        "aws:cloudformation:logical-id": "ManagedNodeGroup",
        "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z"
    }
```

```
}
```

Exemple 3 : pour répertorier toutes les balises d'un profil Amazon EKS Fargate ARNe

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'un ARN de profil Amazon EKS Fargate.

```
aws eks list-tags-for-resource  \
    --resource-arn arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/
my-fargate-profile/d6c76780-e541-0725-c816-36754cab734b
```

Sortie :

```
{
    "tags": {
        "eks-fargate-profile-key-2": "value-2",
        "eks-fargate-profile-key-1": "value-1"
    }
}
```

Exemple 4 : pour répertorier toutes les balises d'un ARN complémentaire Amazon EKS

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'un ARN complémentaire Amazon EKS.

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-
cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f
```

Sortie :

```
{
    "tags": {
        "eks-addon-key-2": "value-2",
        "eks-addon-key-1": "value-1"
    }
}
```

Exemple 5 : pour répertorier toutes les balises associées à l'ARN d'un fournisseur d'identité OIDC Amazon EKS

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées à l'ARN d'un fournisseur d'identité OIDC Amazon EKS.

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-provider/8ac76722-78e4-cec1-ed76-d49eea058622
```

Sortie :

```
{
    "tags": {
        "my-identity-provider": "test"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-update`

L'exemple de code suivant montre comment utiliser`list-update`.

AWS CLI

Exemple 1 : Pour répertorie les mises à jour associées au nom d'un cluster Amazon EKS

L'`list-updates`exemple suivant répertorie toutes les mises à jour IDs pour un nom de cluster Amazon EKS.

```
aws eks list-updates \
    --name my-eks-cluster
```

Sortie :

```
{
    "updateIds": [
        "5f78d14e-c57b-4857-a3e4-cf664ae20949",
        "760e5a3f-adad-48c7-88d3-7ac283c09c26",
        "cd4ec863-bc55-47d5-a377-3971502f529b",
        "f12657ce-e869-4f17-b158-a82ab8b7d937"
```

```
    ]
}
```

Exemple 2 : pour répertorier toutes les mises à jour IDs pour un groupe de nœuds Amazon EKS

L'`list-updates`exemple suivant répertorie toutes les mises à jour IDs pour un groupe de nœuds Amazon EKS.

```
aws eks list-updates \
    --name my-eks-cluster \
    --nodegroup-name my-eks-managed-node-group
```

Sortie :

```
{
    "updateIds": [
        "8c6c1bef-61fe-42ac-a242-89412387b8e7"
    ]
}
```

Exemple 3 : pour répertorier toutes les mises à jour IDs sur un module complémentaire Amazon EKS

L'`list-updates`exemple suivant répertorie toutes les mises à jour IDs d'un module complémentaire Amazon EKS.

```
aws eks list-updates \
    --name my-eks-cluster \
    --addon-name vpc-cni
```

Sortie :

```
{
    "updateIds": [
        "9cdba8d4-79fb-3c83-afe8-00b508d33268"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListUpdateà la section Référence des AWS CLI commandes.

`list-updates`

L'exemple de code suivant montre comment utiliser`list-updates`.

AWS CLI

Pour répertorier les mises à jour d'un cluster

Cet exemple de commande répertorie les mises à jour en cours pour un cluster nommé `example` dans votre région par défaut.

Commande :

```
aws eks list-updates --name example
```

Sortie :

```
{
    "updateIds": [
        "10bddb13-a71b-425a-b0a6-71cd03e59161"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListUpdatesà la section Référence des AWS CLI commandes.

`register-cluster`

L'exemple de code suivant montre comment utiliser`register-cluster`.

AWS CLI

Exemple 1 : enregistrer un cluster Kubernetes EKS_ANYWHERE externe sur Amazon EKS

L'`register-cluster`exemple suivant enregistre un cluster Kubernetes EKS_ANYWHERE externe sur Amazon EKS.

```
aws eks register-cluster \
    --name my-eks-anywhere-cluster \
    --connector-config 'roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider=EKS_ANYWHERE'
```

Sortie :

```
{
    "cluster": {
        "name": "my-eks-anywhere-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
        "createdAt": "2024-04-12T12:38:37.561000-04:00",
        "status": "PENDING",
        "tags": {},
        "connectorConfig": {
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
            "provider": "EKS_ANYWHERE",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

Pour plus d'informations, consultez la section Connexion d'un cluster externe dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : enregistrer un cluster Kubernetes externe sur Amazon EKS

L'`register-cluster`exemple suivant enregistre un cluster Kubernetes EKS_ANYWHERE externe sur Amazon EKS.

```
aws eks register-cluster \
    --name my-eks-anywhere-cluster \
    --connector-config 'roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider=OTHER'
```

Sortie :

```
{
    "cluster": {
        "name": "my-onprem-k8s-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-onprem-k8s-cluster",
        "createdAt": "2024-04-12T12:42:10.861000-04:00",
        "status": "PENDING",
        "tags": {},
        "connectorConfig": {
```

```
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": "2024-04-15T12:42:10.339000-04:00",
            "provider": "OTHER",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

Pour plus d'informations, consultez la section Connexion d'un cluster externe dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous RegisterClusterà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Exemple 1 : pour ajouter les balises spécifiées à un cluster Amazon EKS

L'`tag-resource`exemple suivant ajoute les balises spécifiées à un cluster Amazon EKS.

```
aws eks tag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \
    --tag 'my-eks-cluster-test-1=test-value-1,my-eks-cluster-dev-1=dev-value-2'
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour ajouter les balises spécifiées à un groupe de nœuds Amazon EKS

L'`tag-resource`exemple suivant ajoute les balises spécifiées à un groupe de nœuds Amazon EKS.

```
aws eks tag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \
    --tag 'my-eks-nodegroup-test-1=test-value-1,my-eks-nodegroup-dev-1=dev-value-2'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Exemple 1 : pour supprimer les balises spécifiées d'un cluster Amazon EKS

L'`untag-resource`exemple suivant supprime les balises spécifiées d'un cluster Amazon EKS.

```
aws eks untag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \
    --tag-keys "my-eks-cluster-test-1" "my-eks-cluster-dev-1"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour supprimer les balises spécifiées d'un groupe de nœuds Amazon EKS

L'`untag-resource`exemple suivant supprime les balises spécifiées d'un groupe de nœuds Amazon EKS.

```
aws eks untag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \
    --tag-keys "my-eks-nodegroup-test-1" "my-eks-nodegroup-dev-1"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-access-entry`

L'exemple de code suivant montre comment utiliser`update-access-entry`.

AWS CLI

Pour mettre à jour une entrée d'accès pour un cluster EKS

Ce qui suit `update-access-entry` met à jour une entrée d'accès pour le cluster EKS en ajoutant le groupe Kubernetes. `tester`

```
aws eks update-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --kubernetes-groups tester
```

Sortie :

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:role/Admin",
        "kubernetesGroups": [
            "tester"
        ],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/role/111122223333/Admin/d2cb8183-d6ec-b82a-d967-eca21902a4b4",
        "createdAt": "2025-05-24T11:02:04.432000-05:00",
        "modifiedAt": "2025-05-24T17:08:01.608000-05:00",
        "tags": {},
        "username": "arn:aws:sts::111122223333:assumed-role/Admin/{{SessionName}}",
        "type": "STANDARD"
    }
}
```

Pour plus d'informations, consultez la section Mettre à jour les entrées d'accès dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdateAccessEntryà la section Référence des AWS CLI commandes.

`update-addon`

L'exemple de code suivant montre comment utiliser`update-addon`.

AWS CLI

Exemple 1. Pour mettre à jour un module complémentaire Amazon EKS avec le rôle de compte de service ARN

L'`update-addon`exemple de commande suivant met à jour un module complémentaire Amazon EKS avec l'ARN du rôle de compte de service.

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm
```

Sortie :

```
{
    "update": {
        "id": "c00d2de2-c2e4-3d30-929e-46b8edec2ce4",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm"
            }
        ],
        "updatedAt": "2024-04-12T16:04:55.614000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Mise à jour d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 2. Pour mettre à jour un module complémentaire Amazon EKS avec une version complémentaire spécifique

L'`update-addon`exemple de commande suivant met à jour un module complémentaire Amazon EKS avec une version complémentaire spécifique.

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm \
```

```
    --addon-version v1.16.4-eksbuild.2
```

Sortie :

```
{
    "update": {
        "id": "f58dc0b0-2b18-34bd-bc6a-e4abc0011f36",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.16.4-eksbuild.2"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm"
            }
        ],
        "createdAt": "2024-04-12T16:07:16.550000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Mise à jour d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 3. Pour mettre à jour un module complémentaire Amazon EKS avec des valeurs de configuration personnalisées et résoudre les conflits en détail

L'`update-addon`exemple de commande suivant met à jour un module complémentaire Amazon EKS avec des valeurs de configuration personnalisées et les détails de résolution des conflits.

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values '{"resources": {"limits":{"cpu":"100m"}, "requests":
{"cpu":"50m"}}}' \
```

```
    --resolve-conflicts PRESERVE
```

Sortie :

```
{
    "update": {
        "id": "cd9f2173-a8d8-3004-a90f-032f14326520",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.16.4-eksbuild.2"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "{\"resources\": {\"limits\":{\"cpu\":\"100m\"}, \"requests
\":{\"cpu\":\"50m\"}}}"
            }
        ],
        "createdAt": "2024-04-12T16:16:27.363000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Mise à jour d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 4. Pour mettre à jour un module complémentaire Amazon EKS avec un fichier de valeurs de configuration JSON personnalisé

L'`update-addon`exemple de commande suivant met à jour un module complémentaire Amazon EKS avec des valeurs de configuration JSON personnalisées et les détails de résolution des conflits.

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.17.1-eksbuild.1 \
    --configuration-values 'file://configuration-values.json' \
    --resolve-conflicts PRESERVE
```

Contenu de `configuration-values.json` :

```
{
    "resources": {
        "limits": {
            "cpu": "100m"
        },
        "requests": {
            "cpu": "50m"
        }
    },
    "env": {
        "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
    }
}
```

Sortie :

```
{
    "update": {
        "id": "6881a437-174f-346b-9a63-6e91763507cc",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.17.1-eksbuild.1"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm"
            },
```

```
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "{\n    \"resources\": {\n        \"limits\": {\n
  \"cpu\": \"100m\"\n        },\n        \"requests\": {\n            \"cpu\": \"50m
\"\n        }\n    },\n    \"env\": {\n        \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR
\"\n    }\n}"
            }
        ],
        "createdAt": "2024-04-12T16:22:55.519000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Mise à jour d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

Exemple 5. Pour mettre à jour un module complémentaire Amazon EKS avec un fichier de valeurs de configuration YAML personnalisé

L'`update-addon`exemple de commande suivant met à jour un module complémentaire Amazon EKS avec des valeurs de configuration YAML personnalisées et les détails de résolution des conflits.

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.18.0-eksbuild.1 \
    --configuration-values 'file://configuration-values.yaml' \
    --resolve-conflicts PRESERVE
```

Contenu de `configuration-values.yaml` :

```
resources:
    limits:
        cpu: '100m'
    requests:
```

```
        cpu: '50m'
env:
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

Sortie :

```
{
    "update": {
        "id": "a067a4c9-69d0-3769-ace9-d235c5b16701",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.18.0-eksbuild.1"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "resources:\n    limits:\n        cpu: '100m'\n
 requests:\n        cpu: '50m'\nenv:\n    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
            }
        ],
        "createdAt": "2024-04-12T16:25:07.212000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Gestion des modules complémentaires Amazon EKS - Mise à jour d'un module complémentaire dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdateAddonà la section Référence des AWS CLI commandes.

`update-cluster-config`

L'exemple de code suivant montre comment utiliser`update-cluster-config`.

AWS CLI

Pour mettre à jour l'accès aux terminaux du cluster

Cet exemple de commande met à jour un cluster pour désactiver l'accès public aux points de terminaison et activer l'accès aux points de terminaison privés.

Commande :

```
aws eks update-cluster-config --name example \
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

Sortie :

```
{
    "update": {
        "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",
        "status": "InProgress",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "false"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            }
        ],
        "createdAt": 1565806986.506,
        "errors": []
    }
}
```

Pour activer la journalisation pour un cluster

Cet exemple de commande active tous les types de journalisation du plan de contrôle du cluster pour un cluster nomméexample.

Commande :

```
aws eks update-cluster-config --name example \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Sortie :

```
{
    "update": {
        "id": "7551c64b-1d27-4b1e-9f8e-c45f056eb6fd",
        "status": "InProgress",
        "type": "LoggingUpdate",
        "params": [
            {
                "type": "ClusterLogging",
                "value": "{\"clusterLogging\":[{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}]}"
            }
        ],
        "createdAt": 1565807210.37,
        "errors": []
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateClusterConfigà la section Référence des AWS CLI commandes.

`update-cluster-version`

L'exemple de code suivant montre comment utiliser`update-cluster-version`.

AWS CLI

Pour mettre à jour un cluster Amazon EKS nommé `my-eks-cluster` vers la version Kubernetes spécifiée

L'`update-cluster-version`exemple suivant met à jour un cluster Amazon EKS vers la version Kubernetes spécifiée.

```
aws eks update-cluster-version \
    --name my-eks-cluster \
```

```
    --kubernetes-version 1.27
```

Sortie :

```
{
    "update": {
        "id": "e4091a28-ea14-48fd-a8c7-975aeb469e8a",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.27"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.16"
            }
        ],
        "createdAt": "2024-04-12T16:56:01.082000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'une version Kubernetes d'un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdateClusterVersionà la section Référence des AWS CLI commandes.

`update-kubeconfig`

L'exemple de code suivant montre comment utiliser`update-kubeconfig`.

AWS CLI

Exemple 1 : configure votre kubectl en créant ou en mettant à jour le kubeconfig afin que vous puissiez vous connecter à un cluster Amazon EKS nommé `` my-eks-cluster

L'`update-kubeconfig`exemple suivant configure votre kubectl en créant ou en mettant à jour le kubeconfig afin que vous puissiez vous connecter à un cluster Amazon EKS nommé. my-eks-cluster

```
aws eks update-kubeconfig \
    --name my-eks-cluster
```

Sortie :

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/
xxx/.kube/config
```

Pour plus d'informations, consultez la section Création ou mise à jour d'un fichier kubeconfig pour un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : configure votre kubectl en créant ou en mettant à jour le kubeconfig (avec l'option role-arn pour assumer un rôle dans l'authentification du cluster) afin que vous puissiez vous connecter à un cluster Amazon EKS nommé `` my-eks-cluster

L'`update-kubeconfig`exemple suivant configure votre kubectl en créant ou en mettant à jour le kubeconfig (avec l'option role-arn pour assumer un rôle dans l'authentification du cluster) afin que vous puissiez vous connecter à un cluster Amazon EKS nommé. my-eks-cluster

```
aws eks update-kubeconfig \
    --name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-
cluster-ServiceRole-j1k7AfTIQtnM
```

Sortie :

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/
xxx/.kube/config
```

Pour plus d'informations, consultez la section Création ou mise à jour d'un fichier kubeconfig pour un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : configure votre kubectl en créant ou en mettant à jour le kubeconfig (avec l'option role-arn pour assumer un rôle dans l'authentification du cluster, ainsi qu'un alias de cluster et un alias d'utilisateur personnalisés) afin que vous puissiez vous connecter à un cluster Amazon EKS nommé `` my-eks-cluster

L'`update-kubeconfig`exemple suivant configure votre kubectl en créant ou en mettant à jour le kubeconfig (avec l'option role-arn pour assumer un rôle dans l'authentification du cluster ainsi

qu'un alias de cluster et un alias d'utilisateur personnalisés) afin que vous puissiez vous connecter à un cluster Amazon EKS nommé. my-eks-cluster

```
aws eks update-kubeconfig \
    --name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-ServiceRole-j1k7AfTIQtnM \
    --alias stage-eks-cluster \
    --user-alias john
```

Sortie :

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

Pour plus d'informations, consultez la section Création ou mise à jour d'un fichier kubeconfig pour un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

Exemple 4 : Imprimez les entrées du fichier kubeconfig pour les consulter et configurez votre kubectl afin que vous puissiez vous connecter à un cluster Amazon EKS nommé `` my-eks-cluster

L'`update-kubeconfig`exemple suivant configure votre kubectl en créant ou en mettant à jour le kubeconfig (avec l'option role-arn pour assumer un rôle dans l'authentification du cluster ainsi qu'un alias de cluster et un alias d'utilisateur personnalisés) afin que vous puissiez vous connecter à un cluster Amazon EKS nommé. my-eks-cluster

```
aws eks update-kubeconfig \
    --name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-ServiceRole-j1k7AfTIQtnM \
    --alias stage-eks-cluster \
    --user-alias john \
    --verbose
```

Sortie :

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
Entries:

context:
cluster: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

```
user: john
name: stage-eks-cluster

name: john
user:
exec:
    apiVersion: client.authentication.k8s.io/v1beta1
    args:
    - --region
    - us-east-2
    - eks
    - get-token
    - --cluster-name
    - my-eks-cluster
    - --output
    - json
    - --role
    - arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-
ServiceRole-j1k7AfTIQtnM
    command: aws

cluster:
certificate-authority-data: xxx_CA_DATA_xxx
server: https://DALSJ343KE23J3RN45653DSKJTT647TYD.yl4.us-east-2.eks.amazonaws.com
name: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

Pour plus d'informations, consultez la section Création ou mise à jour d'un fichier kubeconfig pour un cluster Amazon EKS dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdateKubeconfigà la section Référence des AWS CLI commandes.

`update-nodegroup-config`

L'exemple de code suivant montre comment utiliser`update-nodegroup-config`.

AWS CLI

Exemple 1 : mettre à jour un groupe de nœuds gérés pour ajouter de nouvelles étiquettes et altérer le nœud de travail EKS d'un cluster Amazon EKS

L'`update-nodegroup-config`exemple suivant met à jour un groupe de nœuds gérés pour ajouter de nouvelles étiquettes et altérer le nœud de travail EKS d'un cluster Amazon EKS.

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --labels 'addOrUpdateLabels={my-eks-nodegroup-label-1=value-1,my-eks-nodegroup-label-2=value-2}' \
    --taints 'addOrUpdateTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'
```

Sortie :

```
{
    "update": {
        "id": "e66d21d3-bd8b-3ad1-a5aa-b196dc08c7c1",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "LabelsToAdd",
                "value": "{\"my-eks-nodegroup-label-2\":\"value-2\",\"my-eks-nodegroup-label-1\":\"value-1\"}"
            },
            {
                "type": "TaintsToAdd",
                "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",\"key\":\"taint-key-1\"}]"
            }
        ],
        "createdAt": "2024-04-08T12:05:19.161000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : mettre à jour un groupe de nœuds gérés pour supprimer les étiquettes et altérer le nœud de travail EKS d'un cluster Amazon EKS

L'`update-nodegroup-config`exemple suivant met à jour un groupe de nœuds gérés afin de supprimer les étiquettes et d'altérer le nœud de travail EKS d'un cluster Amazon EKS.

```
aws eks update-nodegroup-config \
```

```
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --labels 'removeLabels=my-eks-nodegroup-label-1, my-eks-nodegroup-label-2' \
    --taints 'removeTaints=[{key=taint-key-1,value=taint-
value-1,effect=NO_EXECUTE}]'
```

Sortie :

```
{
    "update": {
        "id": "67a08692-9e59-3ace-a916-13929f44cec3",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "LabelsToRemove",
                "value": "[\"my-eks-nodegroup-label-1\",\"my-eks-nodegroup-
label-2\"]"
            },
            {
                "type": "TaintsToRemove",
                "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",
\"key\":\"taint-key-1\"}]"
            }
        ],
        "createdAt": "2024-04-08T12:17:31.817000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

Exemple 3 : mettre à jour un groupe de nœuds gérés pour supprimer, ajouter des étiquettes et altérer le nœud de travail EKS d'un cluster Amazon EKS

L'`update-nodegroup-config`exemple suivant met à jour un groupe de nœuds gérés afin de supprimer, d'ajouter des étiquettes et d'altérer le nœud de travail EKS d'un cluster Amazon EKS.

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
```

```
    --labels 'addOrUpdateLabels={my-eks-nodegroup-new-label-1=new-value-1,my-eks-
nodegroup-new-label-2=new-value-2},removeLabels=my-eks-nodegroup-label-1, my-eks-
nodegroup-label-2' \
    --taints 'addOrUpdateTaints=[{key=taint-new-key-1,value=taint-new-
value-1,effect=PREFER_NO_SCHEDULE}],removeTaints=[{key=taint-key-1,value=taint-
value-1,effect=NO_EXECUTE}]'
```

Sortie :

```
{
    "update": {
        "id": "4a9c8c45-6ac7-3115-be71-d6412a2339b7",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "LabelsToAdd",
                "value": "{\"my-eks-nodegroup-new-label-1\":\"new-value-1\",\"my-
eks-nodegroup-new-label-2\":\"new-value-2\"}"
            },
            {
                "type": "LabelsToRemove",
                "value": "[\"my-eks-nodegroup-label-1\",\"my-eks-nodegroup-
label-2\"]"
            },
            {
                "type": "TaintsToAdd",
                "value": "[{\"effect\":\"PREFER_NO_SCHEDULE\",\"value\":\"taint-new-
value-1\",\"key\":\"taint-new-key-1\"}]"
            },
            {
                "type": "TaintsToRemove",
                "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",
\"key\":\"taint-key-1\"}]"
            }
        ],
        "createdAt": "2024-04-08T12:30:55.486000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

Exemple 4 : mettre à jour un groupe de nœuds gérés pour mettre à jour scaling-config et update-config pour le nœud de travail EKS d'un cluster Amazon EKS

L'`update-nodegroup-config`exemple suivant met à jour un groupe de nœuds gérés pour mettre à jour scaling-config et update-config pour le nœud de travail EKS d'un cluster Amazon EKS.

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --scaling-config minSize=1,maxSize=5,desiredSize=2 \
    --update-config maxUnavailable=2
```

Sortie :

```
{
    "update": {
        "id": "a977160f-59bf-3023-805d-c9826e460aea",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "MinSize",
                "value": "1"
            },
            {
                "type": "MaxSize",
                "value": "5"
            },
            {
                "type": "DesiredSize",
                "value": "2"
            },
            {
                "type": "MaxUnavailable",
                "value": "2"
            }
        ],
        "createdAt": "2024-04-08T12:35:17.036000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdateNodegroupConfigà la section Référence des AWS CLI commandes.

`update-nodegroup-version`

L'exemple de code suivant montre comment utiliser`update-nodegroup-version`.

AWS CLI

Exemple 1 : mise à jour de la version Kubernetes ou de la version AMI d'un groupe de nœuds géré par Amazon EKS

L'`update-nodegroup-version`exemple suivant met à jour la version Kubernetes ou la version AMI d'un groupe de nœuds géré par Amazon EKS vers la dernière version disponible pour votre cluster Kubernetes.

```
aws eks update-nodegroup-version \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --no-force
```

Sortie :

```
{
    "update": {
        "id": "a94ebfc3-6bf8-307a-89e6-7dbaa36421f7",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.26"
            },
            {
                "type": "ReleaseVersion",
                "value": "1.26.12-20240329"
            }
        ],
        "createdAt": "2024-04-08T13:16:00.724000-04:00",
```

```
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un groupe de nœuds gérés dans le guide de l'utilisateur Amazon EKS.

Exemple 2 : mise à jour de la version Kubernetes ou de la version AMI d'un groupe de nœuds géré par Amazon EKS

L'`update-nodegroup-version`exemple suivant met à jour la version Kubernetes ou la version AMI d'un groupe de nœuds géré par Amazon EKS vers la version de version AMI spécifiée.

```
aws eks update-nodegroup-version \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --kubernetes-version '1.26' \
    --release-version '1.26.12-20240307' \
    --no-force
```

Sortie :

```
{
    "update": {
        "id": "4db06fe1-088d-336b-bdcd-3fdb94995fb7",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.26"
            },
            {
                "type": "ReleaseVersion",
                "value": "1.26.12-20240307"
            }
        ],
        "createdAt": "2024-04-08T13:13:58.595000-04:00",
        "errors": []
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un groupe de nœuds gérés - < https://docs.aws.amazon.com/eks/ latest/userguide/update - managed-node-group .html>` dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdateNodegroupVersionà la section Référence des AWS CLI commandes.

`update-pod-identity-association`

L'exemple de code suivant montre comment utiliser`update-pod-identity-association`.

AWS CLI

Pour mettre à jour l'association EKS Pod Identity

L'`update-pod-identity-association`exemple suivant met à jour une association EKS Pod Identity en remplaçant le rôle IAM associé par « ID `arn:aws:iam::111122223333:role/my-role` `arn:aws:iam::111122223333:role/s3-role` `a-9njjin9gfghecgocd` d'association ». Cette API permet uniquement de mettre à jour le rôle IAM associé.

```
aws eks update-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd \
    --role-arn arn:aws:iam::111122223333:role/s3-role
```

Sortie :

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/s3-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
```

```
        "modifiedAt": "2025-05-25T21:01:53.120000-05:00"
    }
}
```

Pour plus d'informations, consultez la section Découvrez comment EKS Pod Identity permet aux pods d'accéder aux AWS services dans le guide de l'utilisateur Amazon EKS.

- Pour plus de détails sur l'API, reportez-vous UpdatePodIdentityAssociationà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Elastic Beanstalk AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Elastic Beanstalk.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`abort-environment-update`

L'exemple de code suivant montre comment utiliser`abort-environment-update`.

AWS CLI

Pour annuler un déploiement

La commande suivante interrompt le déploiement d'une version d'application en cours d'exécution pour un environnement nommé `my-env` :

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- Pour plus de détails sur l'API, reportez-vous AbortEnvironmentUpdateà la section Référence des AWS CLI commandes.

`check-dns-availability`

L'exemple de code suivant montre comment utiliser`check-dns-availability`.

AWS CLI

Pour vérifier la disponibilité d'un CNAME

La commande suivante vérifie la disponibilité du sous-domaine : `my-cname.elasticbeanstalk.com`

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

Sortie :

```
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- Pour plus de détails sur l'API, reportez-vous CheckDnsAvailabilityà la section Référence des AWS CLI commandes.

`create-application-version`

L'exemple de code suivant montre comment utiliser`create-application-version`.

AWS CLI

Pour créer une nouvelle version de l'application

La commande suivante crée une nouvelle version, « v1 », d'une application nommée MyApp « » :

```
aws elasticbeanstalk create-application-version --application-name MyApp --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="sample.war" --auto-create-application
```

L'application sera créée automatiquement si elle n'existe pas déjà, grâce à auto-create-application cette option. Le bundle source est un fichier .war stocké dans un compartiment s3 nommé « amzn-s3-demo-bucket » qui contient l'exemple d'application Apache Tomcat.

Sortie :

```
{
  "ApplicationVersion": {
      "ApplicationName": "MyApp",
      "VersionLabel": "v1",
      "Description": "MyAppv1",
      "DateCreated": "2015-02-03T23:01:25.412Z",
      "DateUpdated": "2015-02-03T23:01:25.412Z",
      "SourceBundle": {
          "S3Bucket": "amzn-s3-demo-bucket",
          "S3Key": "sample.war"
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateApplicationVersionà la section Référence des AWS CLI commandes.

`create-application`

L'exemple de code suivant montre comment utiliser`create-application`.

AWS CLI

Pour créer une nouvelle application

La commande suivante crée une nouvelle application nommée MyApp « » :

```
aws elasticbeanstalk create-application --application-name MyApp --description "my
 application"
```

La `create-application` commande configure uniquement le nom et la description de l'application. Pour télécharger le code source de l'application, créez une version initiale de l'application à l'aide de`create-application-version`. `create-application-version`dispose également d'une `auto-create-application` option qui vous permet de créer l'application et la version de l'application en une seule étape.

Sortie :

```
{
  "Application": {
      "ApplicationName": "MyApp",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-02-12T18:32:21.181Z",
      "Description": "my application",
      "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateApplicationà la section Référence des AWS CLI commandes.

`create-configuration-template`

L'exemple de code suivant montre comment utiliser`create-configuration-template`.

AWS CLI

Pour créer un modèle de configuration

La commande suivante crée un modèle de configuration nommé `my-app-v1` à partir des paramètres appliqués à un environnement portant l'identifiant `e-rpqsewtp2j` :

```
aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

Sortie :

```
{
    "ApplicationName": "my-app",
    "TemplateName": "my-app-v1",
    "DateCreated": "2015-08-12T18:40:39Z",
    "DateUpdated": "2015-08-12T18:40:39Z",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateConfigurationTemplateà la section Référence des AWS CLI commandes.

`create-environment`

L'exemple de code suivant montre comment utiliser`create-environment`.

AWS CLI

Pour créer un nouvel environnement pour une application

La commande suivante crée un nouvel environnement pour la version « v1 » d'une application Java nommée « my-app » :

```
aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --cname-prefix my-app --version-label v1 --solution-stack-name "64bit
 Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
```

Sortie :

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v1",
  "Status": "Launching",
  "EnvironmentId": "e-izqpassy4h",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
  "CNAME": "my-app.elasticbeanstalk.com",
  "Health": "Grey",
  "Tier": {
      "Type": "Standard",
      "Name": "WebServer",
      "Version": " "
  },
  "DateUpdated": "2015-02-03T23:04:54.479Z",
  "DateCreated": "2015-02-03T23:04:54.479Z"
}
```

`v1`est l'étiquette d'une version d'application précédemment téléchargée avec create-application-version.

Pour spécifier un fichier JSON afin de définir les options de configuration de l'environnement

La `create-environment` commande suivante indique qu'un fichier JSON portant le nom `myoptions.json` doit être utilisé pour remplacer les valeurs obtenues à partir de la pile de solutions ou du modèle de configuration :

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-name sampleapp --option-settings file://myoptions.json
```

`myoptions.json`est un objet JSON définissant plusieurs paramètres :

```
[
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Interval",
    "Value": "15"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Timeout",
    "Value": "8"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "HealthyThreshold",
    "Value": "2"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "UnhealthyThreshold",
    "Value": "3"
  }
]
```

Pour plus d'informations, consultez la section Valeurs des options dans le manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateEnvironmentà la section Référence des AWS CLI commandes.

`create-storage-location`

L'exemple de code suivant montre comment utiliser`create-storage-location`.

AWS CLI

Pour créer un emplacement de stockage

La commande suivante crée un emplacement de stockage dans Amazon S3 :

```
aws elasticbeanstalk create-storage-location
```

Sortie :

```
{
    "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateStorageLocationà la section Référence des AWS CLI commandes.

`delete-application-version`

L'exemple de code suivant montre comment utiliser`delete-application-version`.

AWS CLI

Pour supprimer une version de l'application

La commande suivante supprime une version d'application nommée `22a0-stage-150819_182129` d'après une application nommée `my-app` :

```
aws elasticbeanstalk delete-application-version --version-label 22a0-
stage-150819_182129 --application-name my-app
```

- Pour plus de détails sur l'API, reportez-vous DeleteApplicationVersionà la section Référence des AWS CLI commandes.

`delete-application`

L'exemple de code suivant montre comment utiliser`delete-application`.

AWS CLI

Pour supprimer une application

La commande suivante supprime une application nommée `my-app` :

```
aws elasticbeanstalk delete-application --application-name my-app
```

- Pour plus de détails sur l'API, reportez-vous DeleteApplicationà la section Référence des AWS CLI commandes.

`delete-configuration-template`

L'exemple de code suivant montre comment utiliser`delete-configuration-template`.

AWS CLI

Pour supprimer un modèle de configuration

La commande suivante supprime un modèle de configuration nommé `my-template` d'après une application nommée `my-app` :

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --
application-name my-app
```

- Pour plus de détails sur l'API, reportez-vous DeleteConfigurationTemplateà la section Référence des AWS CLI commandes.

`delete-environment-configuration`

L'exemple de code suivant montre comment utiliser`delete-environment-configuration`.

AWS CLI

Pour supprimer un brouillon de configuration

La commande suivante supprime un brouillon de configuration pour un environnement nommé `my-env` :

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --
application-name my-app
```

- Pour plus de détails sur l'API, reportez-vous DeleteEnvironmentConfigurationà la section Référence des AWS CLI commandes.

`describe-application-versions`

L'exemple de code suivant montre comment utiliser`describe-application-versions`.

AWS CLI

Pour afficher les informations relatives à la version d'une application

La commande suivante permet de récupérer les informations relatives à une version d'application intitulée v2 :

```
aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label "v2"
```

Sortie :

```
{
    "ApplicationVersions": [
        {
            "ApplicationName": "my-app",
            "VersionLabel": "v2",
            "Description": "update cover page",
            "DateCreated": "2015-07-23T01:32:26.079Z",
            "DateUpdated": "2015-07-23T01:32:26.079Z",
            "SourceBundle": {
                "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",
                "S3Key": "my-app/5026-stage-150723_224258.war"
            }
        },
      {
          "ApplicationName": "my-app",
          "VersionLabel": "v1",
          "Description": "initial version",
          "DateCreated": "2015-07-23T22:26:10.816Z",
          "DateUpdated": "2015-07-23T22:26:10.816Z",
          "SourceBundle": {
              "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",
              "S3Key": "my-app/5026-stage-150723_222618.war"
          }
      }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeApplicationVersionsà la section Référence des AWS CLI commandes.

`describe-applications`

L'exemple de code suivant montre comment utiliser`describe-applications`.

AWS CLI

Pour consulter la liste des applications

La commande suivante permet de récupérer des informations sur les applications de la région actuelle :

```
aws elasticbeanstalk describe-applications
```

Sortie :

```
{
    "Applications": [
        {
            "ApplicationName": "ruby",
            "ConfigurationTemplates": [],
            "DateUpdated": "2015-08-13T21:05:44.376Z",
            "Versions": [
                "Sample Application"
            ],
            "DateCreated": "2015-08-13T21:05:44.376Z"
        },
        {
            "ApplicationName": "pythonsample",
            "Description": "Application created from the EB CLI using \"eb init\"",
            "Versions": [
                "Sample Application"
            ],
            "DateCreated": "2015-08-13T19:05:43.637Z",
            "ConfigurationTemplates": [],
            "DateUpdated": "2015-08-13T19:05:43.637Z"
        },
        {
            "ApplicationName": "nodejs-example",
            "ConfigurationTemplates": [],
```

```
            "DateUpdated": "2015-08-06T17:50:02.486Z",
            "Versions": [
                "add elasticache",
                "First Release"
            ],
            "DateCreated": "2015-08-06T17:50:02.486Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeApplicationsà la section Référence des AWS CLI commandes.

`describe-configuration-options`

L'exemple de code suivant montre comment utiliser`describe-configuration-options`.

AWS CLI

Pour afficher les options de configuration d'un environnement

La commande suivante permet de récupérer les descriptions de toutes les options de configuration disponibles pour un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --
application-name my-app
```

Sortie (abrégée) :

```
{
    "Options": [
        {
            "Name": "JVMOptions",
            "UserDefined": false,
            "DefaultValue": "Xms=256m,Xmx=256m,XX:MaxPermSize=64m,JVM Options=",
            "ChangeSeverity": "RestartApplicationServer",
            "Namespace": "aws:cloudformation:template:parameter",
            "ValueType": "KeyValueList"
        },
        {
            "Name": "Interval",
```

```
            "UserDefined": false,
            "DefaultValue": "30",
            "ChangeSeverity": "NoInterruption",
            "Namespace": "aws:elb:healthcheck",
            "MaxValue": 300,
            "MinValue": 5,
            "ValueType": "Scalar"
        },
        ...
        {
            "Name": "LowerThreshold",
            "UserDefined": false,
            "DefaultValue": "2000000",
            "ChangeSeverity": "NoInterruption",
            "Namespace": "aws:autoscaling:trigger",
            "MinValue": 0,
            "ValueType": "Scalar"
        },
        {
            "Name": "ListenerEnabled",
            "UserDefined": false,
            "DefaultValue": "true",
            "ChangeSeverity": "Unknown",
            "Namespace": "aws:elb:listener",
            "ValueType": "Boolean"
        }
    ]
}
```

Les options de configuration disponibles varient en fonction de la plate-forme et de la version de configuration. Pour plus d'informations sur les espaces de noms et les options prises en charge, consultez la section Valeurs des options du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeConfigurationOptionsà la section Référence des AWS CLI commandes.

`describe-configuration-settings`

L'exemple de code suivant montre comment utiliser`describe-configuration-settings`.

AWS CLI

Pour afficher les paramètres de configuration d'un environnement

La commande suivante permet de récupérer les paramètres de configuration d'un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --
application-name my-app
```

Sortie (abrégée) :

```
{
    "ConfigurationSettings": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Description": "Environment created from the EB CLI using \"eb create
\"",
            "DeploymentStatus": "deployed",
            "DateCreated": "2015-08-13T19:16:25Z",
            "OptionSettings": [
                {
                    "OptionName": "Availability Zones",
                    "ResourceName": "AWSEBAutoScalingGroup",
                    "Namespace": "aws:autoscaling:asg",
                    "Value": "Any"
                },
                {
                    "OptionName": "Cooldown",
                    "ResourceName": "AWSEBAutoScalingGroup",
                    "Namespace": "aws:autoscaling:asg",
                    "Value": "360"
                },
                ...
                {
                    "OptionName": "ConnectionDrainingTimeout",
                    "ResourceName": "AWSEBLoadBalancer",
                    "Namespace": "aws:elb:policies",
                    "Value": "20"
                },
                {
                    "OptionName": "ConnectionSettingIdleTimeout",
                    "ResourceName": "AWSEBLoadBalancer",
                    "Namespace": "aws:elb:policies",
                    "Value": "60"
                }
```

```
            ],
            "DateUpdated": "2015-08-13T23:30:07Z",
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
 Java 8"
        }
    ]
}
```

Pour plus d'informations sur les espaces de noms et les options prises en charge, consultez la section Valeurs des options du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeConfigurationSettingsà la section Référence des AWS CLI commandes.

`describe-environment-health`

L'exemple de code suivant montre comment utiliser`describe-environment-health`.

AWS CLI

Pour voir la santé de l'environnement

La commande suivante permet de récupérer les informations générales relatives à l'état de santé d'un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --
attribute-names All
```

Sortie :

```
{
    "Status": "Ready",
    "EnvironmentName": "my-env",
    "Color": "Green",
    "ApplicationMetrics": {
        "Duration": 10,
        "Latency": {
            "P99": 0.004,
            "P75": 0.002,
            "P90": 0.003,
            "P95": 0.004,
            "P85": 0.003,
            "P10": 0.001,
```

```
            "P999": 0.004,
            "P50": 0.001
        },
        "RequestCount": 45,
        "StatusCodes": {
            "Status3xx": 0,
            "Status2xx": 45,
            "Status5xx": 0,
            "Status4xx": 0
        }
    },
    "RefreshedAt": "2015-08-20T21:09:18Z",
    "HealthStatus": "Ok",
    "InstancesHealth": {
        "Info": 0,
        "Ok": 1,
        "Unknown": 0,
        "Severe": 0,
        "Warning": 0,
        "Degraded": 0,
        "NoData": 0,
        "Pending": 0
    },
    "Causes": []
}
```

Les informations de santé ne sont disponibles que pour les environnements dans lesquels les rapports de santé améliorés sont activés. Pour plus d'informations, consultez la section Enhanced Health Reporting and Monitoring du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEnvironmentHealthà la section Référence des AWS CLI commandes.

`describe-environment-resources`

L'exemple de code suivant montre comment utiliser`describe-environment-resources`.

AWS CLI

Pour consulter les informations relatives aux AWS ressources de votre environnement

La commande suivante permet de récupérer des informations sur les ressources d'un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```

Sortie :

```
{
    "EnvironmentResources": {
        "EnvironmentName": "my-env",
        "AutoScalingGroups": [
            {
                "Name": "awseb-e-qu3fyyjyjs-stack-AWSEBAutoScalingGroup-
QSB2ZO88SXZT"
            }
        ],
        "Triggers": [],
        "LoadBalancers": [
            {
                "Name": "awseb-e-q-AWSEBLoa-1EEPZ0K98BIF0"
            }
        ],
        "Queues": [],
        "Instances": [
            {
                "Id": "i-0c91c786"
            }
        ],
        "LaunchConfigurations": [
            {
                "Name": "awseb-e-qu3fyyjyjs-stack-
AWSEBAutoScalingLaunchConfiguration-1UUVQIBC96TQ2"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEnvironmentResourcesà la section Référence des AWS CLI commandes.

`describe-environments`

L'exemple de code suivant montre comment utiliser`describe-environments`.

AWS CLI

Pour afficher les informations relatives à un environnement

La commande suivante permet de récupérer des informations sur un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

Sortie :

```
{
    "Environments": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "VersionLabel": "7f58-stage-150812_025409",
            "Status": "Ready",
            "EnvironmentId": "e-rpqsewtp2j",
            "EndpointURL": "awseb-e-w-AWSEBLoa-1483140XB0Q4L-109QXY8121.us-
west-2.elb.amazonaws.com",
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
 Java 8",
            "CNAME": "my-env.elasticbeanstalk.com",
            "Health": "Green",
            "AbortableOperationInProgress": false,
            "Tier": {
                "Version": " ",
                "Type": "Standard",
                "Name": "WebServer"
            },
            "DateUpdated": "2015-08-12T18:16:55.019Z",
            "DateCreated": "2015-08-07T20:48:49.599Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEnvironmentsà la section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour afficher les événements relatifs à un environnement

La commande suivante récupère les événements d'un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-events --environment-name my-env
```

Sortie (abrégée) :

```
{
    "Events": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Message": "Environment health has transitioned from Info to Ok.",
            "EventDate": "2015-08-20T07:06:53.535Z",
            "Severity": "INFO"
        },
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "b7f3960b-4709-11e5-ba1e-07e16200da41",
            "Message": "Environment update completed successfully.",
            "EventDate": "2015-08-20T07:06:02.049Z"
        },
        ...
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "ca8dfbf6-41ef-11e5-988b-651aa638f46b",
            "Message": "Using elasticbeanstalk-us-west-2-012445113685 as Amazon S3
 storage bucket for environment data.",
            "EventDate": "2015-08-13T19:16:27.561Z"
        },
        {
            "ApplicationName": "my-app",
```

```
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "cdfba8f6-41ef-11e5-988b-65638f41aa6b",
            "Message": "createEnvironment is starting.",
            "EventDate": "2015-08-13T19:16:26.581Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEventsà la section Référence des AWS CLI commandes.

`describe-instances-health`

L'exemple de code suivant montre comment utiliser`describe-instances-health`.

AWS CLI

Pour voir la santé de l'environnement

La commande suivante permet de récupérer les informations relatives à l'état des instances d'un environnement nommé `my-env` :

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --
attribute-names All
```

Sortie :

```
{
    "InstanceHealthList": [
        {
            "InstanceId": "i-08691cc7",
            "ApplicationMetrics": {
                "Duration": 10,
                "Latency": {
                    "P99": 0.006,
                    "P75": 0.002,
                    "P90": 0.004,
                    "P95": 0.005,
                    "P85": 0.003,
                    "P10": 0.0,
```

```
                    "P999": 0.006,
                    "P50": 0.001
                },
                "RequestCount": 48,
                "StatusCodes": {
                    "Status3xx": 0,
                    "Status2xx": 47,
                    "Status5xx": 0,
                    "Status4xx": 1
                }
            },
            "System": {
                "LoadAverage": [
                    0.0,
                    0.02,
                    0.05
                ],
                "CPUUtilization": {
                    "SoftIRQ": 0.1,
                    "IOWait": 0.2,
                    "System": 0.3,
                    "Idle": 97.8,
                    "User": 1.5,
                    "IRQ": 0.0,
                    "Nice": 0.1
                }
            },
            "Color": "Green",
            "HealthStatus": "Ok",
            "LaunchedAt": "2015-08-13T19:17:09Z",
            "Causes": []
        }
    ],
    "RefreshedAt": "2015-08-20T21:09:08Z"
}
```


Les informations de santé ne sont disponibles que pour les environnements dans lesquels les rapports de santé améliorés sont activés. Pour plus d'informations, consultez la section Enhanced Health Reporting and Monitoring du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeInstancesHealthà la section Référence des AWS CLI commandes.

`list-available-solution-stacks`

L'exemple de code suivant montre comment utiliser`list-available-solution-stacks`.

AWS CLI

Pour afficher les piles de solutions

La commande suivante répertorie les piles de solutions pour toutes les configurations de plate-forme actuellement disponibles et celles que vous avez utilisées dans le passé :

```
aws elasticbeanstalk list-available-solution-stacks
```

Sortie (abrégée) :

```
{
    "SolutionStacks": [
        "64bit Amazon Linux 2015.03 v2.0.0 running Node.js",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.6",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.5",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.4",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python 3.4",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python 2.7",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 1.9.3",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 7",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 6",
        "64bit Windows Server Core 2012 R2 running IIS 8.5",
        "64bit Windows Server 2012 R2 running IIS 8.5",
        "64bit Windows Server 2012 running IIS 8",
        "64bit Windows Server 2008 R2 running IIS 7.5",
        "64bit Amazon Linux 2015.03 v2.0.0 running Docker 1.6.2",
        "64bit Amazon Linux 2015.03 v2.0.0 running Multi-container Docker 1.6.2
 (Generic)",
        "64bit Debian jessie v2.0.0 running GlassFish 4.1 Java 8 (Preconfigured -
 Docker)",
```

```
        "64bit Debian jessie v2.0.0 running GlassFish 4.0 Java 7 (Preconfigured -
 Docker)",
        "64bit Debian jessie v2.0.0 running Go 1.4 (Preconfigured - Docker)",
        "64bit Debian jessie v2.0.0 running Go 1.3 (Preconfigured - Docker)",
        "64bit Debian jessie v2.0.0 running Python 3.4 (Preconfigured - Docker)",
    ],
    "SolutionStackDetails": [
        {
            "PermittedFileTypes": [
                "zip"
            ],
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Node.js"
        },
        ...
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAvailableSolutionStacksà la section Référence des AWS CLI commandes.

`rebuild-environment`

L'exemple de code suivant montre comment utiliser`rebuild-environment`.

AWS CLI

Pour reconstruire un environnement

La commande suivante arrête et recrée les ressources dans un environnement nommé : `my-env`

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- Pour plus de détails sur l'API, reportez-vous RebuildEnvironmentà la section Référence des AWS CLI commandes.

`request-environment-info`

L'exemple de code suivant montre comment utiliser`request-environment-info`.

AWS CLI

Pour demander des journaux détaillés

La commande suivante demande des journaux à un environnement nommé `my-env` :

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-type tail
```

Après avoir demandé les journaux, récupérez leur emplacement avec retrieve-environment-info.

- Pour plus de détails sur l'API, reportez-vous RequestEnvironmentInfoà la section Référence des AWS CLI commandes.

`restart-app-server`

L'exemple de code suivant montre comment utiliser`restart-app-server`.

AWS CLI

Pour redémarrer les serveurs d'applications

La commande suivante redémarre les serveurs d'applications sur toutes les instances d'un environnement nommé `my-env` :

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- Pour plus de détails sur l'API, reportez-vous RestartAppServerà la section Référence des AWS CLI commandes.

`retrieve-environment-info`

L'exemple de code suivant montre comment utiliser`retrieve-environment-info`.

AWS CLI

Pour récupérer les journaux suivis

La commande suivante permet de récupérer un lien vers les journaux d'un environnement nommé `my-env` :

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-type tail
```

Sortie :

```
{
    "EnvironmentInfo": [
        {
            "SampleTimestamp": "2015-08-20T22:23:17.703Z",
            "Message": "https://elasticbeanstalk-us-
west-2-0123456789012.s3.amazonaws.com/resources/environments/
logs/tail/e-fyqyju3yjs/i-09c1c867/TailLogs-1440109397703.out?
AWSAccessKeyId=AKGPT4J56IAJ2EUBL5CQ&Expires=1440195891&Signature=n
%2BEalOV6A2HIOx4Rcfb7LT16bBM%3D",
            "InfoType": "tail",
            "Ec2InstanceId": "i-09c1c867"
        }
    ]
}
```

Affichez le lien dans un navigateur. Avant de les récupérer, les journaux doivent être demandés auprès request-environment-info de.

- Pour plus de détails sur l'API, reportez-vous RetrieveEnvironmentInfoà la section Référence des AWS CLI commandes.

`swap-environment-cnames`

L'exemple de code suivant montre comment utiliser`swap-environment-cnames`.

AWS CLI

Pour échanger les CNAMES de l'environnement

La commande suivante échange les sous-domaines assignés à deux environnements :

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue
 --destination-environment-name my-env-green
```

- Pour plus de détails sur l'API, reportez-vous SwapEnvironmentCnamesà la section Référence des AWS CLI commandes.

`terminate-environment`

L'exemple de code suivant montre comment utiliser`terminate-environment`.

AWS CLI

Pour mettre fin à un environnement

La commande suivante met fin à un environnement Elastic Beanstalk nommé : `my-env`

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

Sortie :

```
{
    "ApplicationName": "my-app",
    "EnvironmentName": "my-env",
    "Status": "Terminating",
    "EnvironmentId": "e-fh2eravpns",
    "EndpointURL": "awseb-e-f-AWSEBLoa-1I9XUMP4-8492WNUP202574.us-
west-2.elb.amazonaws.com",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
 8",
    "CNAME": "my-env.elasticbeanstalk.com",
    "Health": "Grey",
    "AbortableOperationInProgress": false,
    "Tier": {
        "Version": " ",
        "Type": "Standard",
        "Name": "WebServer"
    },
    "DateUpdated": "2015-08-12T19:05:54.744Z",
    "DateCreated": "2015-08-12T18:52:53.622Z"
}
```

- Pour plus de détails sur l'API, reportez-vous TerminateEnvironmentà la section Référence des AWS CLI commandes.

`update-application-version`

L'exemple de code suivant montre comment utiliser`update-application-version`.

AWS CLI

Pour modifier la description de la version d'une application

La commande suivante met à jour la description d'une version d'application nommée `22a0-stage-150819_185942` :

```
aws elasticbeanstalk update-application-version --version-label 22a0-stage-150819_185942 --application-name my-app --description "new description"
```

Sortie :

```
{
    "ApplicationVersion": {
        "ApplicationName": "my-app",
        "VersionLabel": "22a0-stage-150819_185942",
        "Description": "new description",
        "DateCreated": "2015-08-19T18:59:17.646Z",
        "DateUpdated": "2015-08-20T22:53:28.871Z",
        "SourceBundle": {
            "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012",
            "S3Key": "my-app/22a0-stage-150819_185942.war"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateApplicationVersionà la section Référence des AWS CLI commandes.

`update-application`

L'exemple de code suivant montre comment utiliser`update-application`.

AWS CLI

Pour modifier la description d'une application

La commande suivante met à jour la description d'une application nommée `my-app` :

```
aws elasticbeanstalk update-application --application-name my-app --description "my Elastic Beanstalk application"
```

Sortie :

```
{
```

```
    "Application": {
        "ApplicationName": "my-app",
        "Description": "my Elastic Beanstalk application",
        "Versions": [
            "2fba-stage-150819_234450",
            "bf07-stage-150820_214945",
            "93f8",
            "fd7c-stage-150820_000431",
            "22a0-stage-150819_185942"
        ],
        "DateCreated": "2015-08-13T19:15:50.449Z",
        "ConfigurationTemplates": [],
        "DateUpdated": "2015-08-20T22:34:56.195Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateApplicationà la section Référence des AWS CLI commandes.

`update-configuration-template`

L'exemple de code suivant montre comment utiliser`update-configuration-template`.

AWS CLI

Pour mettre à jour un modèle de configuration

La commande suivante supprime la configuration des métriques de santé CloudWatch personnalisées configurée `ConfigDocument` d'un modèle de configuration enregistré nommé `my-template` :

```
aws elasticbeanstalk update-configuration-template --template-
name my-template --application-name my-app --options-to-
remove Namespace=aws:elasticbeanstalk:healthreporting:system,OptionName=ConfigDocument
```

Sortie :

```
{
    "ApplicationName": "my-app",
    "TemplateName": "my-template",
    "DateCreated": "2015-08-20T22:39:31Z",
    "DateUpdated": "2015-08-20T22:43:11Z",
```

```
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
}
```

Pour plus d'informations sur les espaces de noms et les options prises en charge, consultez la section Valeurs des options du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateConfigurationTemplateà la section Référence des AWS CLI commandes.

`update-environment`

L'exemple de code suivant montre comment utiliser`update-environment`.

AWS CLI

Pour mettre à jour un environnement vers une nouvelle version

La commande suivante met à jour un environnement nommé « my-env » vers la version « v2 » de l'application à laquelle il appartient :

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

Cette commande nécessite que l'environnement « my-env » existe déjà et appartienne à une application dont la version d'application est valide et porte l'étiquette « v2 ».

Sortie :

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v2",
  "Status": "Updating",
  "EnvironmentId": "e-szqipays4h",
  "EndpointURL": "awseb-e-i-AWSEBLoa-1RDLX6TC9VUAO-0123456789.us-
west-2.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux running Tomcat 7",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "Tier": {
      "Version": " ",
      "Type": "Standard",
      "Name": "WebServer"
```

```
  },
  "DateUpdated": "2015-02-03T23:12:29.119Z",
  "DateCreated": "2015-02-03T23:04:54.453Z"
}
```

Pour définir une variable d'environnement

La commande suivante définit la valeur de la variable « PARAM1 » dans l'environnement « my-env » sur « » : ParamValue

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para
```

Le `option-settings` paramètre prend un espace de noms en plus du nom et de la valeur de la variable. Elastic Beanstalk prend en charge plusieurs espaces de noms pour les options en plus des variables d'environnement.

Pour configurer les paramètres des options à partir d'un fichier

La commande suivante configure plusieurs options dans l'espace de `aws:elb:loadbalancer` noms à partir d'un fichier :

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://options.json
```

`options.json`est un objet JSON définissant plusieurs paramètres :

```
[
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Interval",
    "Value": "15"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Timeout",
    "Value": "8"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "HealthyThreshold",
```

```
    "Value": "2"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "UnhealthyThreshold",
    "Value": "3"
  }
]
```

Sortie :

```
{
    "ApplicationName": "my-app",
    "EnvironmentName": "my-env",
    "VersionLabel": "7f58-stage-150812_025409",
    "Status": "Updating",
    "EnvironmentId": "e-wtp2rpqsej",
    "EndpointURL": "awseb-e-w-AWSEBLoa-14XB83101Q4L-104QXY80921.sa-
east-1.elb.amazonaws.com",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
 8",
    "CNAME": "my-env.elasticbeanstalk.com",
    "Health": "Grey",
    "AbortableOperationInProgress": true,
    "Tier": {
        "Version": " ",
        "Type": "Standard",
        "Name": "WebServer"
    },
    "DateUpdated": "2015-08-12T18:15:23.804Z",
    "DateCreated": "2015-08-07T20:48:49.599Z"
}
```

Pour plus d'informations sur les espaces de noms et les options prises en charge, consultez la section Valeurs des options du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateEnvironmentà la section Référence des AWS CLI commandes.

`validate-configuration-settings`

L'exemple de code suivant montre comment utiliser`validate-configuration-settings`.

AWS CLI

Pour valider les paramètres de configuration

La commande suivante valide un document de configuration de métriques CloudWatch personnalisé :

```
aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings file://options.json
```

`options.json`est un document JSON qui inclut un ou plusieurs paramètres de configuration à valider :

```
[
    {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "ConfigDocument",
        "Value": "{\"CloudWatchMetrics\": {\"Environment\":
 {\"ApplicationLatencyP99.9\": null,\"InstancesSevere\": 60,
\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": null,
\"ApplicationLatencyP95\": 60,\"InstancesUnknown\": 60,\"ApplicationLatencyP85\":
 60,\"InstancesInfo\": null,\"ApplicationRequests2xx\": null,\"InstancesDegraded
\": null,\"InstancesWarning\": 60,\"ApplicationLatencyP50\": 60,
\"ApplicationRequestsTotal\": null,\"InstancesNoData\": null,\"InstancesPending
\": 60,\"ApplicationLatencyP10\": null,\"ApplicationRequests5xx\": null,
\"ApplicationLatencyP75\": null,\"InstancesOk\": 60,\"ApplicationRequests3xx\":
 null,\"ApplicationRequests4xx\": null},\"Instance\": {\"ApplicationLatencyP99.9\":
 null,\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": null,
\"ApplicationLatencyP95\": null,\"ApplicationLatencyP85\": null,\"CPUUser\": 60,
\"ApplicationRequests2xx\": null,\"CPUIdle\": null,\"ApplicationLatencyP50\":
 null,\"ApplicationRequestsTotal\": 60,\"RootFilesystemUtil\": null,
\"LoadAverage1min\": null,\"CPUIrq\": null,\"CPUNice\": 60,\"CPUIowait\": 60,
\"ApplicationLatencyP10\": null,\"LoadAverage5min\": null,\"ApplicationRequests5xx
\": null,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,\"ApplicationRequests3xx\":
 60,\"ApplicationRequests4xx\": null,\"InstanceHealth\": null,\"CPUSoftirq\": 60}},
\"Version\": 1}"
    }
]
```

Si les options que vous spécifiez sont valides pour l'environnement spécifié, Elastic Beanstalk renvoie un tableau de messages vide :

```
{
    "Messages": []
}
```

Si la validation échoue, la réponse inclura des informations sur l'erreur :

```
{
    "Messages": [
        {
            "OptionName": "ConfigDocumet",
            "Message": "Invalid option specification (Namespace:
 'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
 Unknown configuration setting.",
            "Namespace": "aws:elasticbeanstalk:healthreporting:system",
            "Severity": "error"
        }
    ]
}
```

Pour plus d'informations sur les espaces de noms et les options prises en charge, consultez la section Valeurs des options du manuel AWS Elastic Beanstalk Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ValidateConfigurationSettingsà la section Référence des AWS CLI commandes.

Elastic Load Balancing - Version 1 : exemples d'utilisation AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de la AWS Command Line Interface version 1 d'Elastic Load Balancing.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags`

L'exemple de code suivant montre comment utiliser`add-tags`.

AWS CLI

Pour ajouter une balise à un équilibreur de charge

Cet exemple ajoute des balises à l'équilibreur de charge spécifié.

Commande :

```
aws elb add-tags --load-balancer-name my-load-balancer --
tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

- Pour plus de détails sur l'API, reportez-vous AddTagsà la section Référence des AWS CLI commandes.

`apply-security-groups-to-load-balancer`

L'exemple de code suivant montre comment utiliser`apply-security-groups-to-load-balancer`.

AWS CLI

Pour associer un groupe de sécurité à un équilibreur de charge dans un VPC

Cet exemple associe un groupe de sécurité à l'équilibreur de charge spécifié dans un VPC.

Commande :

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer
 --security-groups sg-fc448899
```

Sortie :

```
{
```

```
  "SecurityGroups": [
      "sg-fc448899"
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ApplySecurityGroupsToLoadBalancerà la section Référence des AWS CLI commandes.

`attach-load-balancer-to-subnets`

L'exemple de code suivant montre comment utiliser`attach-load-balancer-to-subnets`.

AWS CLI

Pour associer des sous-réseaux à un équilibreur de charge

Cet exemple ajoute le sous-réseau spécifié à l'ensemble de sous-réseaux configurés pour l'équilibreur de charge spécifié.

Commande :

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

Sortie :

```
{
   "Subnets": [
       "subnet-15aaab61",
       "subnet-0ecac448"
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous AttachLoadBalancerToSubnetsà la section Référence des AWS CLI commandes.

`configure-health-check`

L'exemple de code suivant montre comment utiliser`configure-health-check`.

AWS CLI

Pour spécifier les paramètres de vérification de l'état de vos instances principales EC2

Cet exemple indique les paramètres de vérification de l'état utilisés pour évaluer l'état de santé de vos EC2 instances principales.

Commande :

```
aws elb configure-health-check --load-balancer-name my-load-balancer --health-check Target=HTTP:80/png,Interval=30,UnhealthyThreshold=2,HealthyThreshold=2,Timeout=3
```

Sortie :

```
{
   "HealthCheck": {
       "HealthyThreshold": 2,
       "Interval": 30,
       "Target": "HTTP:80/png",
       "Timeout": 3,
       "UnhealthyThreshold": 2
   }
}
```

- Pour plus de détails sur l'API, reportez-vous ConfigureHealthCheckà la section Référence des AWS CLI commandes.

`create-app-cookie-stickiness-policy`

L'exemple de code suivant montre comment utiliser`create-app-cookie-stickiness-policy`.

AWS CLI

Pour générer une politique de fidélisation pour votre équilibreur de charge HTTPS

Cet exemple génère une politique de persistance qui suit les durées de vie de session persistantes du cookie généré par l'application.

Commande :

```
aws elb create-app-cookie-stickiness-policy --load-balancer-name my-load-balancer --
policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- Pour plus de détails sur l'API, reportez-vous CreateAppCookieStickinessPolicyà la section Référence des AWS CLI commandes.

`create-lb-cookie-stickiness-policy`

L'exemple de code suivant montre comment utiliser`create-lb-cookie-stickiness-policy`.

AWS CLI

Pour générer une politique d'adhérence basée sur la durée pour votre équilibreur de charge HTTPS

Cet exemple génère une politique de fidélisation dont la durée de vie des sessions est contrôlée par la période d'expiration spécifiée.

Commande :

```
aws elb create-lb-cookie-stickiness-policy --load-balancer-name my-load-balancer --
policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- Pour plus de détails sur l'API, reportez-vous CreateLbCookieStickinessPolicyà la section Référence des AWS CLI commandes.

`create-load-balancer-listeners`

L'exemple de code suivant montre comment utiliser`create-load-balancer-listeners`.

AWS CLI

Pour créer des écouteurs HTTP pour un équilibreur de charge

Cet exemple crée un écouteur pour votre équilibreur de charge sur le port 80 à l'aide du protocole HTTP.

Commande :

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

Pour créer des écouteurs HTTPS pour un équilibreur de charge

Cet exemple crée un écouteur pour votre équilibreur de charge sur le port 443 à l'aide du protocole HTTPS.

Commande :

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- Pour plus de détails sur l'API, reportez-vous CreateLoadBalancerListenersà la section Référence des AWS CLI commandes.

`create-load-balancer-policy`

L'exemple de code suivant montre comment utiliser`create-load-balancer-policy`.

AWS CLI

Pour créer une politique qui active le protocole proxy sur un équilibreur de charge

Cet exemple crée une politique qui active le protocole proxy sur l'équilibreur de charge spécifié.

Commande :

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-
name my-ProxyProtocol-policy --policy-type-name ProxyProtocolPolicyType --policy-
attributes AttributeName=ProxyProtocol,AttributeValue=true
```

Pour créer une politique de négociation SSL à l'aide de la politique de sécurité recommandée

Cet exemple crée une politique de négociation SSL pour l'équilibreur de charge HTTPS spécifié en utilisant la politique de sécurité recommandée.

Commande :

```
aws elb create-load-balancer-policy --load-balancer-name my-load-
balancer --policy-name my-SSLNegotiation-policy --policy-type-
name SSLNegotiationPolicyType --policy-attributes AttributeName=Reference-Security-
Policy,AttributeValue=ELBSecurityPolicy-2015-03
```

Pour créer une politique de négociation SSL à l'aide d'une politique de sécurité personnalisée

Cet exemple crée une politique de négociation SSL pour votre équilibreur de charge HTTPS à l'aide d'une politique de sécurité personnalisée en activant les protocoles et les chiffrements.

Commande :

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-
name my-SSLNegotiation-policy --policy-type-name SSLNegotiationPolicyType --policy-
attributes AttributeName=Protocol-SSLv3,AttributeValue=true AttributeName=Protocol-
TLSv1.1,AttributeValue=true AttributeName=DHE-RSA-AES256-
SHA256,AttributeValue=true AttributeName=Server-Defined-Cipher-
Order,AttributeValue=true
```

Pour créer une politique de clé publique

Cet exemple crée une politique de clé publique.

Commande :

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --
policy-name my-PublicKey-policy --policy-type-name PublicKeyPolicyType --policy-
attributes AttributeName=PublicKey,AttributeValue=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ
+dS74kj//c6x7ROtusUaeQCTgIUkayttRDWchuqo1pHC1u
+n5xxXnBBe2ejbb2WRsKIQ5rXEeixsjFpFsojpSQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe
+o4haAWvDBALJU0pkSI1FecPHcs2hwxf14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCEOjnYRg7RPvybaGfQ
+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPUbcUkkXDj45vC2s3Z8E
+Lk7a3uZhvsQHLZnrfuWjBWGWvZ/MhZYgEXAMPLE
```

Pour créer une politique d'authentification du serveur principal

Cet exemple crée une politique d'authentification du serveur principal qui permet l'authentification sur votre instance principale à l'aide d'une politique de clé publique.

Commande :

```
aws elb create-load-balancer-policy --load-balancer-name my-
load-balancer --policy-name my-authentication-policy --policy-
type-name BackendServerAuthenticationPolicyType --policy-
attributes AttributeName=PublicKeyPolicyName,AttributeValue=my-PublicKey-policy
```

- Pour plus de détails sur l'API, reportez-vous CreateLoadBalancerPolicyà la section Référence des AWS CLI commandes.

`create-load-balancer`

L'exemple de code suivant montre comment utiliser`create-load-balancer`.

AWS CLI

Pour créer un équilibreur de charge HTTP

Cet exemple crée un équilibreur de charge avec un écouteur HTTP dans un VPC.

Commande :

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

Sortie :

```
{
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

Cet exemple crée un équilibreur de charge avec un écouteur HTTP dans EC2 -Classic.

Commande :

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --availability-zones us-west-2a us-west-2b
```

Sortie :

```
{
    "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

Pour créer un équilibreur de charge HTTPS

Cet exemple crée un équilibreur de charge avec un écouteur HTTPS dans un VPC.

Commande :

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protoco
certificate/my-server-cert" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

Sortie :

```
{
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

Cet exemple crée un équilibreur de charge avec un écouteur HTTPS dans EC2 -Classic.

Commande :

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protoco
certificate/my-server-cert" --availability-zones us-west-2a us-west-2b
```

Sortie :

```
{
    "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

Pour créer un équilibreur de charge interne

Cet exemple crée un équilibreur de charge interne avec un écouteur HTTP dans un VPC.

Commande :

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --scheme internal --subnets subnet-a85db0df --security-groups sg-a61988c3
```

Sortie :

```
{
    "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLoadBalancerà la section Référence des AWS CLI commandes.

`delete-load-balancer-listeners`

L'exemple de code suivant montre comment utiliser`delete-load-balancer-listeners`.

AWS CLI

Pour supprimer un écouteur de votre équilibreur de charge

Cet exemple supprime l'écouteur du port spécifié de l'équilibreur de charge spécifié.

Commande :

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-balancer-ports 80
```

- Pour plus de détails sur l'API, reportez-vous DeleteLoadBalancerListenersà la section Référence des AWS CLI commandes.

`delete-load-balancer-policy`

L'exemple de code suivant montre comment utiliser`delete-load-balancer-policy`.

AWS CLI

Pour supprimer une politique de votre équilibreur de charge

Cet exemple supprime la politique spécifiée de l'équilibreur de charge spécifié. La politique ne doit être activée sur aucun écouteur.

Commande :

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- Pour plus de détails sur l'API, reportez-vous DeleteLoadBalancerPolicyà la section Référence des AWS CLI commandes.

`delete-load-balancer`

L'exemple de code suivant montre comment utiliser`delete-load-balancer`.

AWS CLI

Pour supprimer un équilibreur de charge

Cet exemple supprime l'équilibreur de charge spécifié.

Commande :

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- Pour plus de détails sur l'API, reportez-vous DeleteLoadBalancerà la section Référence des AWS CLI commandes.

`deregister-instances-from-load-balancer`

L'exemple de code suivant montre comment utiliser`deregister-instances-from-load-balancer`.

AWS CLI

Pour désenregistrer des instances d'un équilibreur de charge

Cet exemple désenregistre l'instance spécifiée de l'équilibreur de charge spécifié.

Commande :

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-balancer --instances i-d6f6fae3
```

Sortie :

```
{
    "Instances": [
```

```
        {
            "InstanceId": "i-207d9717"
        },
        {
            "InstanceId": "i-afefb49b"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeregisterInstancesFromLoadBalancerà la section Référence des AWS CLI commandes.

`describe-account-limits`

L'exemple de code suivant montre comment utiliser`describe-account-limits`.

AWS CLI

Pour décrire les limites de votre Classic Load Balancer

L'`describe-account-limits`exemple suivant affiche des informations sur les limites du Classic Load Balancer pour votre AWS compte.

```
aws elb describe-account-limits
```

Sortie :

```
{
    "Limits": [
        {
            "Name": "classic-load-balancers",
            "Max": "20"
        },
        {
            "Name": "classic-listeners",
            "Max": "100"
        },
        {
            "Name": "classic-registered-instances",
            "Max": "1000"
        }
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAccountLimitsà la section Référence des AWS CLI commandes.

`describe-instance-health`

L'exemple de code suivant montre comment utiliser`describe-instance-health`.

AWS CLI

Pour décrire l'état des instances d'un équilibreur de charge

Cet exemple décrit l'état des instances de l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

Sortie :

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-207d9717",
          "ReasonCode": "N/A",
          "State": "InService",
          "Description": "N/A"
      },
      {
          "InstanceId": "i-afefb49b",
          "ReasonCode": "N/A",
          "State": "InService",
          "Description": "N/A"
      }
  ]
}
```

Pour décrire l'état d'une instance d'un équilibreur de charge

Cet exemple décrit l'état de santé de l'instance spécifiée pour l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --
instances i-7299c809
```

Voici un exemple de réponse pour une instance en cours d'enregistrement.

Sortie :

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "ELB",
          "State": "OutOfService",
          "Description": "Instance registration is still in progress."
    }
  ]
}
```

Voici un exemple de réponse pour une instance défectueuse.

Sortie :

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "Instance",
          "State": "OutOfService",
          "Description": "Instance has failed at least the UnhealthyThreshold number
 of health checks consecutively."
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceHealthà la section Référence des AWS CLI commandes.

`describe-load-balancer-attributes`

L'exemple de code suivant montre comment utiliser`describe-load-balancer-attributes`.

AWS CLI

Pour décrire les attributs d'un équilibreur de charge

Cet exemple décrit les attributs de l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

Sortie :

```
{
  "LoadBalancerAttributes": {
      "ConnectionDraining": {
          "Enabled": false,
          "Timeout": 300
      },
      "CrossZoneLoadBalancing": {
          "Enabled": true
      },
      "ConnectionSettings": {
          "IdleTimeout": 30
      },
      "AccessLog": {
          "Enabled": false
    }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancerAttributesà la section Référence des AWS CLI commandes.

`describe-load-balancer-policies`

L'exemple de code suivant montre comment utiliser`describe-load-balancer-policies`.

AWS CLI

Pour décrire toutes les politiques associées à un équilibreur de charge

Cet exemple décrit toutes les politiques associées à l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

Sortie :

```
{
  "PolicyDescriptions": [
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "ProxyProtocol",
          "AttributeValue": "true"
        }
      ],
      "PolicyName": "my-ProxyProtocol-policy",
      "PolicyTypeName": "ProxyProtocolPolicyType"
    },
    {
        "PolicyAttributeDescriptions": [
            {
                "AttributeName": "CookieName",
                "AttributeValue": "my-app-cookie"
            }
        ],
        "PolicyName": "my-app-cookie-policy",
        "PolicyTypeName": "AppCookieStickinessPolicyType"
    },
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "CookieExpirationPeriod",
          "AttributeValue": "60"
        }
      ],
      "PolicyName": "my-duration-cookie-policy",
      "PolicyTypeName": "LBCookieStickinessPolicyType"
    },
    .
    .
    .
  ]
}
```

Pour décrire une politique spécifique associée à un équilibreur de charge

Cet exemple décrit la politique spécifiée associée à l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --
policy-name my-authentication-policy
```

Sortie :

```
{
  "PolicyDescriptions": [
      {
          "PolicyAttributeDescriptions": [
              {
                  "AttributeName": "PublicKeyPolicyName",
                  "AttributeValue": "my-PublicKey-policy"
              }
          ],
          "PolicyName": "my-authentication-policy",
          "PolicyTypeName": "BackendServerAuthenticationPolicyType"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancerPoliciesà la section Référence des AWS CLI commandes.

`describe-load-balancer-policy-types`

L'exemple de code suivant montre comment utiliser`describe-load-balancer-policy-types`.

AWS CLI

Pour décrire les types de politique d'équilibreur de charge définis par Elastic Load Balancing

Cet exemple décrit les types de politique d'équilibreur de charge que vous pouvez utiliser pour créer des configurations de politique pour votre équilibreur de charge.

Commande :

```
aws elb describe-load-balancer-policy-types
```

Sortie :

```
{
  "PolicyTypeDescriptions": [
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE",
                  "AttributeName": "ProxyProtocol",
                  "AttributeType": "Boolean"
              }
          ],
          "PolicyTypeName": "ProxyProtocolPolicyType",
          "Description": "Policy that controls whether to include the IP address and
 port of the originating request for TCP messages. This policy operates on TCP/SSL
 listeners only"
      },
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE",
                  "AttributeName": "PublicKey",
                  "AttributeType": "String"
              }
          ],
          "PolicyTypeName": "PublicKeyPolicyType",
          "Description": "Policy containing a list of public keys to
 accept when authenticating the back-end server(s). This policy cannot be
 applied directly to back-end servers or listeners but must be part of a
 BackendServerAuthenticationPolicyType."
      },
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE",
                  "AttributeName": "CookieName",
                  "AttributeType": "String"
              }
          ],
          "PolicyTypeName": "AppCookieStickinessPolicyType",
```

```
          "Description": "Stickiness policy with session lifetimes controlled by the
 lifetime of the application-generated cookie. This policy can be associated only
 with HTTP/HTTPS listeners."
      },
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ZERO_OR_ONE",
                  "AttributeName": "CookieExpirationPeriod",
                  "AttributeType": "Long"
              }
          ],
          "PolicyTypeName": "LBCookieStickinessPolicyType",
          "Description": "Stickiness policy with session lifetimes controlled by
 the browser (user-agent) or a specified expiration period. This policy can be
 associated only with HTTP/HTTPS listeners."
      },
      {
          "PolicyAttributeTypeDescriptions": [
              .
              .
              .
          ],
          "PolicyTypeName": "SSLNegotiationPolicyType",
          "Description": "Listener policy that defines the ciphers and protocols
 that will be accepted by the load balancer. This policy can be associated only with
 HTTPS/SSL listeners."
      },
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE_OR_MORE",
                  "AttributeName": "PublicKeyPolicyName",
                  "AttributeType": "PolicyName"
              }
          ],
          "PolicyTypeName": "BackendServerAuthenticationPolicyType",
          "Description": "Policy that controls authentication to back-end server(s)
 and contains one or more policies, such as an instance of a PublicKeyPolicyType.
 This policy can be associated only with back-end servers that are using HTTPS/SSL."
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancerPolicyTypesà la section Référence des AWS CLI commandes.

`describe-load-balancers`

L'exemple de code suivant montre comment utiliser`describe-load-balancers`.

AWS CLI

Pour décrire vos équilibreurs de charge

Cet exemple décrit tous vos équilibreurs de charge.

Commande :

```
aws elb describe-load-balancers
```

Pour décrire l'un de vos équilibreurs de charge

Cet exemple décrit l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```

L'exemple de réponse suivant concerne un équilibreur de charge HTTPS dans un VPC.

Sortie :

```
{
  "LoadBalancerDescriptions": [
    {
      "Subnets": [
          "subnet-15aaab61"
      ],
      "CanonicalHostedZoneNameID": "Z3DZXE0EXAMPLE",
      "CanonicalHostedZoneName": "my-load-balancer-1234567890.us-
west-2.elb.amazonaws.com",
      "ListenerDescriptions": [
          {
              "Listener": {
                  "InstancePort": 80,
                  "LoadBalancerPort": 80,
```

```
                "Protocol": "HTTP",
                "InstanceProtocol": "HTTP"
            },
            "PolicyNames": []
        },
        {
            "Listener": {
                "InstancePort": 443,
                "SSLCertificateId": "arn:aws:iam::123456789012:server-certificate/
my-server-cert",
                "LoadBalancerPort": 443,
                "Protocol": "HTTPS",
                "InstanceProtocol": "HTTPS"
            },
            "PolicyNames": [
                "ELBSecurityPolicy-2015-03"
            ]
        }
    ],
    "HealthCheck": {
        "HealthyThreshold": 2,
        "Interval": 30,
        "Target": "HTTP:80/png",
        "Timeout": 3,
        "UnhealthyThreshold": 2
    },
    "VPCId": "vpc-a01106c2",
    "BackendServerDescriptions": [
        {
            "InstancePort": 80,
            "PolicyNames": [
                "my-ProxyProtocol-policy"
            ]
        }
    ],
    "Instances": [
        {
            "InstanceId": "i-207d9717"
        },
        {
            "InstanceId": "i-afefb49b"
        }
    ],
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com",
```

```
      "SecurityGroups": [
          "sg-a61988c3"
      ],
      "Policies": {
          "LBCookieStickinessPolicies": [
              {
                  "PolicyName": "my-duration-cookie-policy",
                  "CookieExpirationPeriod": 60
              }
          ],
          "AppCookieStickinessPolicies": [],
          "OtherPolicies": [
              "my-PublicKey-policy",
              "my-authentication-policy",
              "my-SSLNegotiation-policy",
              "my-ProxyProtocol-policy",
              "ELBSecurityPolicy-2015-03"
          ]
      },
      "LoadBalancerName": "my-load-balancer",
      "CreatedTime": "2015-03-19T03:24:02.650Z",
      "AvailabilityZones": [
          "us-west-2a"
      ],
      "Scheme": "internet-facing",
      "SourceSecurityGroup": {
          "OwnerAlias": "123456789012",
          "GroupName": "my-elb-sg"
      }
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancersà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire les balises attribuées à un équilibreur de charge

Cet exemple décrit les balises attribuées à l'équilibreur de charge spécifié.

Commande :

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

Sortie :

```
{
  "TagDescriptions": [
      {
          "Tags": [
              {
                  "Value": "lima",
                  "Key": "project"
              },
              {
                  "Value": "digital-media",
                  "Key": "department"
              }
          ],
          "LoadBalancerName": "my-load-balancer"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`detach-load-balancer-from-subnets`

L'exemple de code suivant montre comment utiliser`detach-load-balancer-from-subnets`.

AWS CLI

Pour détacher les équilibreurs de charge des sous-réseaux

Cet exemple détache l'équilibreur de charge spécifié du sous-réseau spécifié.

Commande :

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

Sortie :

```
{
   "Subnets": [
       "subnet-15aaab61"
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous DetachLoadBalancerFromSubnetsà la section Référence des AWS CLI commandes.

`disable-availability-zones-for-load-balancer`

L'exemple de code suivant montre comment utiliser`disable-availability-zones-for-load-balancer`.

AWS CLI

Pour désactiver les zones de disponibilité pour un équilibreur de charge

Cet exemple supprime la zone de disponibilité spécifiée de l'ensemble des zones de disponibilité pour l'équilibreur de charge spécifié.

Commande :

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-load-
balancer --availability-zones us-west-2a
```

Sortie :

```
{
    "AvailabilityZones": [
        "us-west-2b"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DisableAvailabilityZonesForLoadBalancerà la section Référence des AWS CLI commandes.

`enable-availability-zones-for-load-balancer`

L'exemple de code suivant montre comment utiliser`enable-availability-zones-for-load-balancer`.

AWS CLI

Pour activer les zones de disponibilité pour un équilibreur de charge

Cet exemple ajoute la zone de disponibilité spécifiée à l'équilibreur de charge spécifié.

Commande :

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2b
```

Sortie :

```
{
    "AvailabilityZones": [
        "us-west-2a",
        "us-west-2b"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous [EnableAvailabilityZonesForLoadBalancer](#)à la section Référence des AWS CLI commandes.

`modify-load-balancer-attributes`

L'exemple de code suivant montre comment utiliser`modify-load-balancer-attributes`.

AWS CLI

Pour modifier les attributs d'un équilibreur de charge

Cet exemple modifie l'`CrossZoneLoadBalancing`attribut de l'équilibreur de charge spécifié.

Commande :

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

Sortie :

```
{
    "LoadBalancerAttributes": {
        "CrossZoneLoadBalancing": {
            "Enabled": true
        }
    },
    "LoadBalancerName": "my-load-balancer"
}
```

Cet exemple modifie l'`ConnectionDraining`attribut de l'équilibreur de charge spécifié.

Commande :

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer
 --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout
\":300}}"
```

Sortie :

```
{
    "LoadBalancerAttributes": {
        "ConnectionDraining": {
            "Enabled": true,
            "Timeout": 300
        }
    },
    "LoadBalancerName": "my-load-balancer"
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyLoadBalancerAttributesà la section Référence des AWS CLI commandes.

`register-instances-with-load-balancer`

L'exemple de code suivant montre comment utiliser`register-instances-with-load-balancer`.

AWS CLI

Pour enregistrer des instances auprès d'un équilibreur de charge

Cet exemple enregistre l'instance spécifiée auprès de l'équilibreur de charge spécifié.

Commande :

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
 --instances i-d6f6fae3
```

Sortie :

```
{
   "Instances": [
       {
           "InstanceId": "i-d6f6fae3"
       },
       {
           "InstanceId": "i-207d9717"
       },
       {
           "InstanceId": "i-afefb49b"
       }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous RegisterInstancesWithLoadBalancerà la section Référence des AWS CLI commandes.

`remove-tags`

L'exemple de code suivant montre comment utiliser`remove-tags`.

AWS CLI

Pour supprimer des balises d'un équilibreur de charge

Cet exemple supprime une balise de l'équilibreur de charge spécifié.

Commande :

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- Pour plus de détails sur l'API, reportez-vous RemoveTagsà la section Référence des AWS CLI commandes.

`set-load-balancer-listener-ssl-certificate`

L'exemple de code suivant montre comment utiliser`set-load-balancer-listener-ssl-certificate`.

AWS CLI

Pour mettre à jour le certificat SSL d'un équilibreur de charge HTTPS

Cet exemple remplace le certificat SSL existant pour l'équilibreur de charge HTTPS spécifié.

Commande :

```
aws elb set-load-balancer-listener-ssl-certificate --load-balancer-
name my-load-balancer --load-balancer-port 443 --ssl-certificate-
id arn:aws:iam::123456789012:server-certificate/new-server-cert
```

- Pour plus de détails sur l'API, reportez-vous SetLoadBalancerListenerSslCertificateà la section Référence des AWS CLI commandes.

`set-load-balancer-policies-for-backend-server`

L'exemple de code suivant montre comment utiliser`set-load-balancer-policies-for-backend-server`.

AWS CLI

Pour remplacer les politiques associées à un port pour une instance principale

Cet exemple remplace les politiques actuellement associées au port spécifié.

Commande :

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-
balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

Pour supprimer toutes les politiques actuellement associées à un port sur votre instance principale

Cet exemple supprime toutes les politiques associées au port spécifié.

Commande :

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names []
```

Pour confirmer que les politiques sont supprimées, utilisez la `describe-load-balancer-policies` commande.

- Pour plus de détails sur l'API, reportez-vous SetLoadBalancerPoliciesForBackendServerà la section Référence des AWS CLI commandes.

`set-load-balancer-policies-of-listener`

L'exemple de code suivant montre comment utiliser`set-load-balancer-policies-of-listener`.

AWS CLI

Pour remplacer les politiques associées à un écouteur

Cet exemple remplace les politiques actuellement associées à l'écouteur spécifié.

Commande :

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
 --load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

Pour supprimer toutes les politiques associées à votre écouteur

Cet exemple supprime toutes les politiques actuellement associées à l'écouteur spécifié.

Commande :

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
 --load-balancer-port 443 --policy-names []
```

Pour confirmer que les politiques sont supprimées de l'équilibreur de charge, utilisez la `describe-load-balancer-policies` commande.

- Pour plus de détails sur l'API, reportez-vous SetLoadBalancerPoliciesOfListenerà la section Référence des AWS CLI commandes.

Elastic Load Balancing - Version 2 : exemples d'utilisation AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de la AWS Command Line Interface version 2 d'Elastic Load Balancing.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-listener-certificates`

L'exemple de code suivant montre comment utiliser`add-listener-certificates`.

AWS CLI

Pour ajouter un certificat à un écouteur sécurisé

Cet exemple ajoute le certificat spécifié à l'écouteur sécurisé spécifié.

Commande :

```
aws elbv2 add-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705
```

Sortie :

```
{
  "Certificates": [
      {
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
```

```
            "IsDefault": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AddListenerCertificatesà la section Référence des AWS CLI commandes.

`add-tags`

L'exemple de code suivant montre comment utiliser`add-tags`.

AWS CLI

Pour ajouter des balises à un équilibreur de charge

L'`add-tags`exemple suivant ajoute les `department` balises `project` et à l'équilibreur de charge spécifié.

```
aws elbv2 add-tags \
    --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

- Pour plus de détails sur l'API, reportez-vous AddTagsà la section Référence des AWS CLI commandes.

`create-listener`

L'exemple de code suivant montre comment utiliser`create-listener`.

AWS CLI

Exemple 1 : pour créer un écouteur HTTP

L'`create-listener`exemple suivant crée un écouteur HTTP pour l'Application Load Balancer spécifié qui transmet les demandes au groupe cible spécifié.

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
```

```
    --protocol HTTP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Pour plus d'informations, consultez Tutoriel : Création d'un équilibreur de charge d'application à l'aide de la AWS CLI dans le guide de l'utilisateur pour les équilibreurs de charge d'application.

Exemple 2 : pour créer un écouteur HTTPS

L'`create-listener`exemple suivant crée un écouteur HTTPS pour l'Application Load Balancer spécifié qui transmet les demandes au groupe cible spécifié. Vous devez spécifier un certificat SSL pour un écouteur HTTPS. Vous pouvez créer et gérer des certificats à l'aide de AWS Certificate Manager (ACM). Vous pouvez également créer un certificat à l'aide d' SSL/TLS outils, le faire signer par une autorité de certification (CA) et le télécharger dans AWS Identity and Access Management (IAM).

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --protocol HTTPS \
    --port 443 \
    --certificates CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
    --ssl-policy ELBSecurityPolicy-2016-08 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Pour plus d'informations, voir Ajouter un écouteur HTTPS dans le guide de l'utilisateur pour les équilibreurs de charge d'application.

Exemple 3 : pour créer un écouteur TCP

L'`create-listener`exemple suivant crée un écouteur TCP pour le Network Load Balancer spécifié qui transmet les demandes au groupe cible spécifié.

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \
    --protocol TCP \
    --port 80 \
```

```
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

Pour plus d'informations, consultez Tutoriel : Création d'un Network Load Balancer à l'aide de la AWS CLI dans le Guide de l'utilisateur pour les Network Load Balancers.

Exemple 4 : pour créer un écouteur TLS

L'`create-listener`exemple suivant crée un écouteur TLS pour le Network Load Balancer spécifié qui transmet les demandes au groupe cible spécifié. Vous devez spécifier un certificat SSL pour un écouteur TLS.

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --protocol TLS \
    --port 443 \
    --certificates CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
    --ssl-policy ELBSecurityPolicy-2016-08 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Pour plus d'informations, consultez la section Écouteurs TLS pour votre Network Load Balancer dans le Guide de l'utilisateur pour les Network Load Balancers.

Exemple 5 : pour créer un écouteur UDP

L'`create-listener`exemple suivant crée un écouteur UDP pour le Network Load Balancer spécifié qui transmet les demandes au groupe cible spécifié.

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \
    --protocol UDP \
    --port 53 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

Pour plus d'informations, consultez Tutoriel : Création d'un Network Load Balancer à l'aide de la AWS CLI dans le Guide de l'utilisateur pour les Network Load Balancers.

Exemple 6 : pour créer un écouteur pour la passerelle et le transfert spécifiés

L'`create-listener`exemple suivant crée un écouteur pour le Gateway Load Balancer spécifié qui transmet les demandes au groupe cible spécifié.

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/e0f9b3d5c7f7d3d6 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/my-glb-targets/007ca469fae3bb1615
```

Sortie :

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:listener/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6/
afc127db15f925de",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6",
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615",
                    "ForwardConfig": {
                        "TargetGroups": [
                            {
                                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with Gateway Load Balancers using the AWS CLI dans le Guide de l'utilisateur des Gateway Load Balancers.

- Pour plus de détails sur l'API, reportez-vous CreateListeneràla section Référence des AWS CLI commandes.

`create-load-balancer`

L'exemple de code suivant montre comment utiliser`create-load-balancer`.

AWS CLI

Exemple 1 : pour créer un équilibreur de charge connecté à Internet

L'`create-load-balancer`exemple suivant crée un Application Load Balancer connecté à Internet et active les zones de disponibilité pour les sous-réseaux spécifiés.

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --subnets subnet-b7d581c0 subnet-8360a9e7
```

Sortie :

```
{
    "LoadBalancers": [
        {
            "Type": "application",
            "Scheme": "internet-facing",
            "IpAddressType": "ipv4",
            "VpcId": "vpc-3ac0fb5f",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-west-2a",
                    "SubnetId": "subnet-8360a9e7"
                },
                {
                    "ZoneName": "us-west-2b",
                    "SubnetId": "subnet-b7d581c0"
                }
            ],
            "CreatedTime": "2017-08-25T21:26:12.920Z",
            "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
            "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
            "SecurityGroups": [
                "sg-5943793c"
```

```
            ],
            "LoadBalancerName": "my-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
        }
    ]
}
```

Pour plus d'informations, consultez Tutoriel : Création d'un équilibreur de charge d'application à l'aide de la AWS CLI dans le guide de l'utilisateur pour les équilibreurs de charge d'application.

Exemple 2 : pour créer un équilibreur de charge interne

L'`create-load-balancer`exemple suivant crée un Application Load Balancer interne et active les zones de disponibilité pour les sous-réseaux spécifiés.

```
aws elbv2 create-load-balancer \
    --name my-internal-load-balancer \
    --scheme internal \
    --subnets subnet-b7d581c0 subnet-8360a9e7
```

Sortie :

```
{
    "LoadBalancers": [
        {
            "Type": "application",
            "Scheme": "internal",
            "IpAddressType": "ipv4",
            "VpcId": "vpc-3ac0fb5f",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-west-2a",
                    "SubnetId": "subnet-8360a9e7"
                },
                {
                    "ZoneName": "us-west-2b",
                    "SubnetId": "subnet-b7d581c0"
                }
            ],
```

```
            "CreatedTime": "2016-03-25T21:29:48.850Z",
            "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
            "DNSName": "internal-my-internal-load-balancer-1529930873.us-
west-2.elb.amazonaws.com",
            "SecurityGroups": [
                "sg-5943793c"
            ],
            "LoadBalancerName": "my-internal-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-internal-load-balancer/5b49b8d4303115c2"
        }
    ]
}
```

Pour plus d'informations, consultez Tutoriel : Création d'un équilibreur de charge d'application à l'aide de la AWS CLI dans le guide de l'utilisateur pour les équilibreurs de charge d'application.

Exemple 3 : pour créer un Network Load Balancer

L'`create-load-balancer`exemple suivant crée un Network Load Balancer connecté à Internet et active la zone de disponibilité pour le sous-réseau spécifié. Il utilise un mappage de sous-réseau pour associer l'adresse IP élastique spécifiée à l'interface réseau utilisée par les nœuds d'équilibrage de charge pour la zone de disponibilité.

```
aws elbv2 create-load-balancer \
    --name my-network-load-balancer \
    --type network \
    --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a
```

Sortie :

```
{
    "LoadBalancers": [
        {
            "Type": "network",
            "Scheme": "internet-facing",
            "IpAddressType": "ipv4",
            "VpcId": "vpc-3ac0fb5f",
            "AvailabilityZones": [
```

```
                {
                    "LoadBalancerAddresses": [
                        {
                            "IpAddress": "35.161.207.171",
                            "AllocationId": "eipalloc-64d5890a"
                        }
                    ],
                    "ZoneName": "us-west-2b",
                    "SubnetId": "subnet-5264e837"
                }
            ],
            "CreatedTime": "2017-10-15T22:41:25.657Z",
            "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
            "DNSName": "my-network-load-balancer-5d1b75f4f1cee11e.elb.us-
west-2.amazonaws.com",
            "LoadBalancerName": "my-network-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e"
        }
    ]
}
```

Pour plus d'informations, consultez Tutoriel : Création d'un Network Load Balancer à l'aide de la AWS CLI dans le Guide de l'utilisateur pour les Network Load Balancers.

Exemple 4 : pour créer un Gateway Load Balancer

L'`create-load-balancer`exemple suivant crée un Gateway Load Balancer et active les zones de disponibilité pour les sous-réseaux spécifiés.

```
aws elbv2 create-load-balancer \
    --name my-gateway-load-balancer \
    --type gateway \
    --subnets subnet-dc83f691 subnet-a62583f9
```

Sortie :

```
{
    "LoadBalancers": [
        {
```

```
            "Type": "gateway",
            "VpcId": "vpc-838475fe",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-1b",
                    "SubnetId": "subnet-a62583f9"
                },
            {
                    "ZoneName": "us-east-1a",
                    "SubnetId": "subnet-dc83f691"
                }
            ],
            "CreatedTime": "2021-07-14T19:33:43.324000+00:00",
            "LoadBalancerName": "my-gateway-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/dfbb5a7d32cdee79"
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with Gateway Load Balancers using the AWS CLI dans le Guide de l'utilisateur des Gateway Load Balancers.

- Pour plus de détails sur l'API, reportez-vous CreateLoadBalancerà la section Référence des AWS CLI commandes.

`create-rule`

L'exemple de code suivant montre comment utiliser`create-rule`.

AWS CLI

Exemple 1 : pour créer une règle à l'aide d'une condition de chemin et d'une action directe

L'`create-rule`exemple suivant crée une règle qui transmet les demandes au groupe cible spécifié si l'URL contient le modèle spécifié.

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
```

```
    --priority 5 \
    --conditions file://conditions-pattern.json
    --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Contenu de `conditions-pattern.json` :

```
[
    {
        "Field": "path-pattern",
        "PathPatternConfig": {
            "Values": ["/images/*"]
        }
    }
]
```

Exemple 2 : pour créer une règle à l'aide d'une condition d'hôte et d'une réponse fixe

L'`create-rule`exemple suivant crée une règle qui fournit une réponse fixe si le nom d'hôte dans l'en-tête d'hôte correspond au nom d'hôte spécifié.

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --priority 10 \
    --conditions file://conditions-host.json \
    --actions file://actions-fixed-response.json
```

Contenu de `conditions-host.json`

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "Values": ["*.example.com"]
      }
  }
]
```

Contenu de `actions-fixed-response.json`

```
[
```

```
    {
        "Type": "fixed-response",
        "FixedResponseConfig": {
            "MessageBody": "Hello world",
            "StatusCode": "200",
            "ContentType": "text/plain"
        }
    }
]
```

Exemple 3 : pour créer une règle à l'aide d'une condition d'adresse IP source, d'une action d'authentification et d'une action de transfert

L'`create-rule`exemple suivant crée une règle qui authentifie l'utilisateur si l'adresse IP source correspond à l'adresse IP spécifiée, et transmet la demande au groupe cible spécifié si l'authentification est réussie.

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --priority 20 \
    --conditions file://conditions-source-ip.json \
    --actions file://actions-authenticate.json
```

Contenu de `conditions-source-ip.json`

```
[
    {
        "Field": "source-ip",
        "SourceIpConfig": {
            "Values": ["192.0.2.0/24", "198.51.100.10/32"]
        }
    }
]
```

Contenu de `actions-authenticate.json`

```
[
    {
        "Type": "authenticate-oidc",
        "AuthenticateOidcConfig": {
            "Issuer": "https://idp-issuer.com",
```

```
            "AuthorizationEndpoint": "https://authorization-endpoint.com",
            "TokenEndpoint": "https://token-endpoint.com",
            "UserInfoEndpoint": "https://user-info-endpoint.com",
            "ClientId": "abcdefghijklmnopqrstuvwxyz123456789",
            "ClientSecret": "1234567890123456789012345678 90",
            "SessionCookieName": "my-cookie",
            "SessionTimeout": 3600,
            "Scope": "email",
            "AuthenticationRequestExtraParams": {
                "display": "page",
                "prompt": "login"
            },
            "OnUnauthenticatedRequest": "deny"
        },
        "Order": 1
    },
    {
        "Type": "forward",
        "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:880185128111:targetgroup/cli-test/642a97ecb0e0f26b",
        "Order": 2
    }
]
```

- Pour plus de détails sur l'API, reportez-vous CreateRuleà la section Référence des AWS CLI commandes.

`create-target-group`

L'exemple de code suivant montre comment utiliser`create-target-group`.

AWS CLI

Exemple 1 : pour créer un groupe cible pour un Application Load Balancer

L'`create-target-group`exemple suivant crée un groupe cible pour un Application Load Balancer dans lequel vous enregistrez les cibles par ID d'instance (le type de cible est`instance`). Ce groupe cible utilise le protocole HTTP, le port 80 et les paramètres de contrôle de santé par défaut d'un groupe cible HTTP.

```
aws elbv2 create-target-group \
    --name my-targets \
```

```
    --protocol HTTP \
    --port 80 \
    --target-type instance \
    --vpc-id vpc-3ac0fb5f
```

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "TargetGroupName": "my-targets",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

Pour plus d'informations, voir Création d'un groupe cible dans le Guide de l'utilisateur pour les équilibreurs de charge d'application.

Exemple 2 : créer un groupe cible pour acheminer le trafic d'un Application Load Balancer vers une fonction Lambda

L'`create-target-group`exemple suivant crée un groupe cible pour un Application Load Balancer dont la cible est une fonction Lambda (le type de cible est). `lambda` Les contrôles de santé sont désactivés par défaut pour ce groupe cible.

```
aws elbv2 create-target-group \
    --name my-lambda-target \
    --target-type lambda
```

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-lambda-target/a3003e085dbb8ddc",
            "TargetGroupName": "my-lambda-target",
            "HealthCheckEnabled": false,
            "HealthCheckIntervalSeconds": 35,
            "HealthCheckTimeoutSeconds": 30,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "lambda",
            "IpAddressType": "ipv4"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Fonctions Lambda en tant que cibles (langue française non garantie) dans le Guide de l'utilisateur pour les Application Load Balancers.

Exemple 3 : pour créer un groupe cible pour un Network Load Balancer

L'`create-target-group`exemple suivant crée un groupe cible pour un Network Load Balancer dans lequel vous enregistrez les cibles par adresse IP (le type de cible est`ip`). Ce groupe cible utilise le protocole TCP, le port 80 et les paramètres de contrôle de santé par défaut d'un groupe cible TCP.

```
aws elbv2 create-target-group \
```

```
    --name my-ip-targets \
    --protocol TCP \
    --port 80 \
    --target-type ip \
    --vpc-id vpc-3ac0fb5f
```

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-ip-targets/b6bba954d1361c78",
            "TargetGroupName": "my-ip-targets",
            "Protocol": "TCP",
            "Port": 80,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckEnabled": true,
            "HealthCheckProtocol": "TCP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 10,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "TargetType": "ip",
            "IpAddressType": "ipv4"
        }
    ]
}
```

Pour plus d'informations, voir Création d'un groupe cible dans le Guide de l'utilisateur pour les équilibreurs de charge réseau.

Exemple 4 : créer un groupe cible pour acheminer le trafic d'un Network Load Balancer vers un Application Load Balancer

L'`create-target-group`exemple suivant crée un groupe cible pour un Network Load Balancer dans lequel vous enregistrez un Application Load Balancer en tant que cible (le type de cible est). `alb`

aws elbv2 create-target-group --name my-alb-target --protocol TCP --port 80 --target-type alb --vpc-id vpc-3ac0fb5f

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-alb-target/a3003e085dbb8ddc",
            "TargetGroupName": "my-alb-target",
            "Protocol": "TCP",
            "Port": 80,
            "VpcId": "vpc-838475fe",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 6,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200-399"
            },
            "TargetType": "alb",
            "IpAddressType": "ipv4"
        }
    ]
}
```

Pour plus d'informations, voir Créer un groupe cible avec un Application Load Balancer comme cible dans le Guide de l'utilisateur pour les Network Load Balancers.

Exemple 5 : pour créer un groupe cible pour un Gateway Load Balancer

L'`create-target-group`exemple suivant crée un groupe cible pour un Gateway Load Balancer dont la cible est une instance et le protocole du groupe cible l'est. `GENEVE`

```
aws elbv2 create-target-group \
    --name my-glb-targetgroup \
    --protocol GENEVE \
    --port 6081 \
    --target-type instance \
    --vpc-id vpc-838475fe
```

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-glb-targetgroup/00c3d57eacd6f40b6f",
            "TargetGroupName": "my-glb-targetgroup",
            "Protocol": "GENEVE",
            "Port": 6081,
            "VpcId": "vpc-838475fe",
            "HealthCheckProtocol": "TCP",
            "HealthCheckPort": "80",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 10,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "TargetType": "instance"
        }
    ]
}
```

Pour plus d'informations, consultez Create a target group < https://docs.aws.amazon.com/elasticloadbalancing/ latest/gateway/create -target-group.html>`__ dans le guide de l'utilisateur de Gateway Load Balancer.

- Pour plus de détails sur l'API, reportez-vous CreateTargetGroupà la section Référence des AWS CLI commandes.

`delete-listener`

L'exemple de code suivant montre comment utiliser`delete-listener`.

AWS CLI

Pour supprimer un écouteur

L'`delete-listener`exemple suivant supprime l'écouteur spécifié.

```
aws elbv2 delete-listener \
```

```
    --listener-arn arn:aws:elasticloadbalancing:ua-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- Pour plus de détails sur l'API, reportez-vous DeleteListenerà la section Référence des AWS CLI commandes.

`delete-load-balancer`

L'exemple de code suivant montre comment utiliser`delete-load-balancer`.

AWS CLI

Pour supprimer un équilibreur de charge

L'`delete-load-balancer`exemple suivant supprime l'équilibreur de charge spécifié.

```
aws elbv2 delete-load-balancer \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- Pour plus de détails sur l'API, reportez-vous DeleteLoadBalancerà la section Référence des AWS CLI commandes.

`delete-rule`

L'exemple de code suivant montre comment utiliser`delete-rule`.

AWS CLI

Pour supprimer une règle

L'`delete-rule`exemple suivant supprime la règle spécifiée.

```
aws elbv2 delete-rule \
    --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- Pour plus de détails sur l'API, reportez-vous DeleteRuleà la section Référence des AWS CLI commandes.

`delete-target-group`

L'exemple de code suivant montre comment utiliser`delete-target-group`.

AWS CLI

Pour supprimer un groupe cible

L'`delete-target-group`exemple suivant supprime le groupe cible spécifié.

```
aws elbv2 delete-target-group \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer un équilibreur de charge dans le guide Application Load Balancer.

- Pour plus de détails sur l'API, reportez-vous DeleteTargetGroupà la section Référence des AWS CLI commandes.

`deregister-targets`

L'exemple de code suivant montre comment utiliser`deregister-targets`.

AWS CLI

Exemple 1 : pour désenregistrer une cible d'un groupe cible

L'`deregister-targets`exemple suivant supprime l'instance spécifiée du groupe cible spécifié.

```
aws elbv2 deregister-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
    --targets Id=i-1234567890abcdef0
```

Exemple 2 : pour désenregistrer une cible enregistrée à l'aide de remplacements de port

L'`deregister-targets`exemple suivant supprime une instance d'un groupe cible qui a été enregistrée à l'aide de remplacements de port.

```
aws elbv2 deregister-targets \
```

```
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \
    --targets Id=i-1234567890abcdef0,Port=80 Id=i-1234567890abcdef0,Port=766
```

- Pour plus de détails sur l'API, reportez-vous DeregisterTargetsà la section Référence des AWS CLI commandes.

`describe-account-limits`

L'exemple de code suivant montre comment utiliser`describe-account-limits`.

AWS CLI

Pour décrire vos limites d'Elastic Load Balancing

L'`describe-account-limits`exemple suivant montre les limites d'Elastic Load Balancing pour votre AWS compte dans la région actuelle.

```
aws elbv2 describe-account-limits
```

Sortie :

```
{
    "Limits": [
        {
            "Name": "target-groups",
            "Max": "3000"
        },
        {
            "Name": "targets-per-application-load-balancer",
            "Max": "1000"
        },
        {
            "Name": "listeners-per-application-load-balancer",
            "Max": "50"
        },
        {
            "Name": "rules-per-application-load-balancer",
            "Max": "100"
        },
        {
            "Name": "network-load-balancers",
```

```
            "Max": "50"
        },
        {
            "Name": "targets-per-network-load-balancer",
            "Max": "3000"
        },
        {
            "Name": "targets-per-availability-zone-per-network-load-balancer",
            "Max": "500"
        },
        {
            "Name": "listeners-per-network-load-balancer",
            "Max": "50"
        },
        {
            "Name": "condition-values-per-alb-rule",
            "Max": "5"
        },
        {
            "Name": "condition-wildcards-per-alb-rule",
            "Max": "5"
        },
        {
            "Name": "target-groups-per-application-load-balancer",
            "Max": "100"
        },
        {
            "Name": "target-groups-per-action-on-application-load-balancer",
            "Max": "5"
        },
        {
            "Name": "target-groups-per-action-on-network-load-balancer",
            "Max": "1"
        },
        {
            "Name": "certificates-per-application-load-balancer",
            "Max": "25"
        },
        {
            "Name": "certificates-per-network-load-balancer",
            "Max": "25"
        },
        {
            "Name": "targets-per-target-group",
```

```
            "Max": "1000"
        },
        {
            "Name": "target-id-registrations-per-application-load-balancer",
            "Max": "1000"
        },
        {
            "Name": "network-load-balancer-enis-per-vpc",
            "Max": "1200"
        },
        {
            "Name": "application-load-balancers",
            "Max": "50"
        },
        {
            "Name": "gateway-load-balancers",
            "Max": "100"
        },
        {
            "Name": "gateway-load-balancers-per-vpc",
            "Max": "100"
        },
        {
            "Name": "geneve-target-groups",
            "Max": "100"
        },
        {
            "Name": "targets-per-availability-zone-per-gateway-load-balancer",
            "Max": "300"
        }
    ]
}
```

Pour plus d'informations, consultez la section Quotas dans le AWS manuel de référence général.

- Pour plus de détails sur l'API, reportez-vous DescribeAccountLimitsà la section Référence des AWS CLI commandes.

`describe-listener-certificates`

L'exemple de code suivant montre comment utiliser`describe-listener-certificates`.

AWS CLI

Pour décrire les certificats d'un écouteur sécurisé

Cet exemple décrit les certificats pour l'écouteur sécurisé spécifié.

Commande :

```
aws elbv2 describe-listener-certificates --listener-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-
balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Sortie :

```
{
  "Certificates": [
      {
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
          "IsDefault": false
      },
      {
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557",
          "IsDefault": false
      },
      {
          "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/
fe59da96-6f58-4a22-8eed-6d0d50477e1d",
          "IsDefault": true
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeListenerCertificatesà la section Référence des AWS CLI commandes.

`describe-listeners`

L'exemple de code suivant montre comment utiliser`describe-listeners`.

AWS CLI

Pour décrire un auditeur

Cet exemple décrit l'écouteur spécifié.

Commande :

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Sortie :

```
{
  "Listeners": [
      {
          "Port": 80,
          "Protocol": "HTTP",
          "DefaultActions": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                  "Type": "forward"
              }
          ],
          "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
          "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
      }
  ]
}
```

Pour décrire les écouteurs d'un équilibreur de charge

Cet exemple décrit les écouteurs de l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Sortie :

```
{
  "Listeners": [
      {
          "Port": 443,
          "Protocol": "HTTPS",
          "DefaultActions": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                  "Type": "forward"
              }
          ],
          "SslPolicy": "ELBSecurityPolicy-2015-05",
          "Certificates": [
              {
                  "CertificateArn": "arn:aws:iam::123456789012:server-certificate/
my-server-cert"
              }
          ],
          "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
          "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65"
      },
      {
          "Port": 80,
          "Protocol": "HTTP",
          "DefaultActions": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                  "Type": "forward"
              }
          ],
          "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
          "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeListenersà la section Référence des AWS CLI commandes.

`describe-load-balancer-attributes`

L'exemple de code suivant montre comment utiliser`describe-load-balancer-attributes`.

AWS CLI

Pour décrire les attributs de l'équilibreur de charge

L'`describe-load-balancer-attributes`exemple suivant affiche les attributs de l'équilibreur de charge spécifié.

```
aws elbv2 describe-load-balancer-attributes \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

L'exemple de sortie suivant montre les attributs d'un Application Load Balancer.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "access_logs.s3.enabled"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.bucket"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.prefix"
        },
        {
            "Value": "60",
            "Key": "idle_timeout.timeout_seconds"
        },
        {
            "Value": "false",
            "Key": "deletion_protection.enabled"
        },
```

```
        {
            "Value": "true",
            "Key": "routing.http2.enabled"
        }
    ]
}
```

L'exemple de sortie suivant inclut les attributs d'un Network Load Balancer.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "access_logs.s3.enabled"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.bucket"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.prefix"
        },
        {
            "Value": "false",
            "Key": "deletion_protection.enabled"
        },
        {
            "Value": "false",
            "Key": "load_balancing.cross_zone.enabled"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancerAttributesà la section Référence des AWS CLI commandes.

`describe-load-balancers`

L'exemple de code suivant montre comment utiliser`describe-load-balancers`.

AWS CLI

Pour décrire un équilibreur de charge

Cet exemple décrit l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 describe-load-balancers --load-balancer-
arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188
```

Sortie :

```
{
  "LoadBalancers": [
      {
          "Type": "application",
          "Scheme": "internet-facing",
          "IpAddressType": "ipv4",
          "VpcId": "vpc-3ac0fb5f",
          "AvailabilityZones": [
              {
                  "ZoneName": "us-west-2a",
                  "SubnetId": "subnet-8360a9e7"
              },
              {
                  "ZoneName": "us-west-2b",
                  "SubnetId": "subnet-b7d581c0"
              }
          ],
          "CreatedTime": "2016-03-25T21:26:12.920Z",
          "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
          "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
          "SecurityGroups": [
              "sg-5943793c"
          ],
          "LoadBalancerName": "my-load-balancer",
          "State": {
              "Code": "active"
          },
          "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
```

```
        }
    ]
}
```

Pour décrire tous les équilibreurs de charge

Cet exemple décrit tous vos équilibreurs de charge.

Commande :

```
aws elbv2 describe-load-balancers
```

- Pour plus de détails sur l'API, reportez-vous DescribeLoadBalancersà la section Référence des AWS CLI commandes.

`describe-rules`

L'exemple de code suivant montre comment utiliser`describe-rules`.

AWS CLI

Exemple 1 : pour décrire une règle

L'`describe-rules`exemple suivant affiche les détails de la règle spécifiée.

```
aws elbv2 describe-rules \
    --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

Exemple 2 : pour décrire les règles applicables à un écouteur

L'`describe-rules`exemple suivant affiche les détails des règles pour l'écouteur spécifié. La sortie inclut la règle par défaut et toutes les autres règles que vous avez ajoutées.

```
aws elbv2 describe-rules \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- Pour plus de détails sur l'API, reportez-vous DescribeRulesà la section Référence des AWS CLI commandes.

`describe-ssl-policies`

L'exemple de code suivant montre comment utiliser`describe-ssl-policies`.

AWS CLI

Exemple 1 : pour répertorier les politiques utilisées pour la négociation SSL par type d'équilibreur de charge

L'`describe-ssl-policies`exemple suivant affiche les noms des politiques que vous pouvez utiliser pour la négociation SSL avec un Application Load Balancer. L'exemple utilise le `--query` paramètre pour afficher uniquement les noms des politiques.

```
aws elbv2 describe-ssl-policies \
    --load-balancer-type application \
    --query SslPolicies[*].Name
```

Sortie :

```
[
    "ELBSecurityPolicy-2016-08",
    "ELBSecurityPolicy-TLS13-1-2-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",
    "ELBSecurityPolicy-TLS13-1-1-2021-06",
    "ELBSecurityPolicy-TLS13-1-0-2021-06",
    "ELBSecurityPolicy-TLS13-1-3-2021-06",
    "ELBSecurityPolicy-TLS-1-2-2017-01",
    "ELBSecurityPolicy-TLS-1-1-2017-01",
    "ELBSecurityPolicy-TLS-1-2-Ext-2018-06",
    "ELBSecurityPolicy-FS-2018-06",
    "ELBSecurityPolicy-2015-05",
    "ELBSecurityPolicy-TLS-1-0-2015-04",
    "ELBSecurityPolicy-FS-1-2-Res-2019-08",
    "ELBSecurityPolicy-FS-1-1-2019-08",
    "ELBSecurityPolicy-FS-1-2-2019-08",
    "ELBSecurityPolicy-FS-1-2-Res-2020-10"
]
```

Exemple 2 : pour répertorier les politiques qui prennent en charge un protocole spécifique

L'`describe-ssl-policies`exemple suivant affiche les noms des politiques qui prennent en charge le protocole TLS 1.3. L'exemple utilise le `--query` paramètre pour afficher uniquement les noms des politiques.

```
aws elbv2 describe-ssl-policies \
    --load-balancer-type application \
    --query SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name
```

Sortie :

```
[
    "ELBSecurityPolicy-TLS13-1-2-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",
    "ELBSecurityPolicy-TLS13-1-1-2021-06",
    "ELBSecurityPolicy-TLS13-1-0-2021-06",
    "ELBSecurityPolicy-TLS13-1-3-2021-06"
]
```

Exemple 3 : pour afficher les chiffrements d'une politique

L'`describe-ssl-policies`exemple suivant affiche les noms des chiffrements pour la politique spécifiée. L'exemple utilise le `--query` paramètre pour afficher uniquement les noms de chiffrement. Le premier chiffre de la liste a la priorité 1, et les autres sont classés par ordre de priorité.

```
aws elbv2 describe-ssl-policies \
    --names ELBSecurityPolicy-TLS13-1-2-2021-06 \
    --query SslPolicies[*].Ciphers[*].Name
```

Sortie :

```
[
    "TLS_AES_128_GCM_SHA256",
    "TLS_AES_256_GCM_SHA384",
    "TLS_CHACHA20_POLY1305_SHA256",
    "ECDHE-ECDSA-AES128-GCM-SHA256",
    "ECDHE-RSA-AES128-GCM-SHA256",
    "ECDHE-ECDSA-AES128-SHA256",
    "ECDHE-RSA-AES128-SHA256",
```

```
    "ECDHE-ECDSA-AES256-GCM-SHA384",
    "ECDHE-RSA-AES256-GCM-SHA384",
    "ECDHE-ECDSA-AES256-SHA384",
    "ECDHE-RSA-AES256-SHA384"
]
```

Pour plus d'informations, consultez la section Politiques de sécurité dans le guide de l'utilisateur pour les équilibreurs de charge d'application.

- Pour plus de détails sur l'API, reportez-vous DescribeSslPoliciesà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire les balises attribuées à un équilibreur de charge

Cet exemple décrit les balises attribuées à l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Sortie :

```
{
  "TagDescriptions": [
      {
          "ResourceArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
          "Tags": [
              {
                  "Value": "lima",
                  "Key": "project"
              },
              {
                  "Value": "digital-media",
                  "Key": "department"
              }
```

```
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`describe-target-group-attributes`

L'exemple de code suivant montre comment utiliser`describe-target-group-attributes`.

AWS CLI

Pour décrire les attributs du groupe cible

L'`describe-target-group-attributes`exemple suivant affiche les attributs du groupe cible spécifié.

```
aws elbv2 describe-target-group-attributes \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

La sortie inclut les attributs si le protocole est HTTP ou HTTPS et si le type de cible est `instance` ou`ip`.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "stickiness.enabled"
        },
        {
            "Value": "300",
            "Key": "deregistration_delay.timeout_seconds"
        },
        {
            "Value": "lb_cookie",
            "Key": "stickiness.type"
        },
        {
            "Value": "86400",
```

```
            "Key": "stickiness.lb_cookie.duration_seconds"
        },
        {
            "Value": "0",
            "Key": "slow_start.duration_seconds"
        }
    ]
}
```

La sortie suivante inclut les attributs si le protocole est HTTP ou HTTPS et si le type de cible est`lambda`.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "lambda.multi_value_headers.enabled"
        }
    ]
}
```

La sortie suivante inclut les attributs si le protocole est TCP, TLS, UDP ou TCP_UDP.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "proxy_protocol_v2.enabled"
        },
        {
            "Value": "300",
            "Key": "deregistration_delay.timeout_seconds"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTargetGroupAttributesà la section Référence des AWS CLI commandes.

`describe-target-groups`

L'exemple de code suivant montre comment utiliser`describe-target-groups`.

AWS CLI

Exemple 1 : pour décrire un groupe cible

L'`describe-target-groups`exemple suivant affiche les détails du groupe cible spécifié.

```
aws elbv2 describe-target-groups \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "TargetGroupName": "my-targets",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "LoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/my-load-balancer/50dc6c495c0c9188"
            ],
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

Exemple 2 : pour décrire tous les groupes cibles d'un équilibreur de charge

L'`describe-target-groups`exemple suivant affiche les détails de tous les groupes cibles pour l'équilibreur de charge spécifié. L'exemple utilise le `--query` paramètre pour afficher uniquement les noms des groupes cibles.

```
aws elbv2 describe-target-groups \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --query TargetGroups[*].TargetGroupName
```

Sortie :

```
[
    "my-instance-targets",
    "my-ip-targets",
    "my-lambda-target"
]
```

Pour plus d'informations, consultez la section Groupes cibles dans le Guide des équilibreurs de charge des applications.

- Pour plus de détails sur l'API, reportez-vous DescribeTargetGroupsà la section Référence des AWS CLI commandes.

`describe-target-health`

L'exemple de code suivant montre comment utiliser`describe-target-health`.

AWS CLI

Exemple 1 : Pour décrire l'état de santé des cibles d'un groupe cible

L'`describe-target-health`exemple suivant affiche les détails de santé des cibles du groupe cible spécifié. Ces cibles sont salutaires.

```
aws elbv2 describe-target-health \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Sortie :

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-ceddcd4d",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        },
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        }
    ]
}
```

Exemple 2 : Pour décrire l'état de santé d'une cible

L'`describe-target-health`exemple suivant affiche les informations relatives à l'état de santé de la cible spécifiée. Cet objectif est salutaire.

```
aws elbv2 describe-target-health \
    --targets Id=i-0f76fade,Port=80 \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Sortie :

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
```

```
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        }
    ]
}
```

L'exemple de sortie suivant concerne une cible dont le groupe cible n'est pas spécifié dans une action pour un écouteur. Cette cible ne peut pas recevoir de trafic en provenance de l'équilibreur de charge.

```
{
    "TargetHealthDescriptions": [
    {
        "HealthCheckPort": "80",
        "Target": {
            "Id": "i-0f76fade",
            "Port": 80
        },
            "TargetHealth": {
                "State": "unused",
                "Reason": "Target.NotInUse",
                "Description": "Target group is not configured to receive traffic
 from the load balancer"
            }
        }
    ]
}
```

L'exemple de sortie suivant concerne une cible dont le groupe cible vient d'être spécifié dans une action pour un écouteur. La cible est toujours en cours d'enregistrement.

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
```

```
            },
            "TargetHealth": {
                "State": "initial",
                "Reason": "Elb.RegistrationInProgress",
                "Description": "Target registration is in progress"
            }
        }
    ]
}
```

L'exemple de sortie suivant concerne une cible défectueuse.

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "unhealthy",
                "Reason": "Target.Timeout",
                "Description": "Connection to target timed out"
            }
        }
    ]
}
```

L'exemple de sortie suivant concerne une cible qui est une fonction Lambda et les contrôles de santé sont désactivés.

```
{
    "TargetHealthDescriptions": [
        {
            "Target": {
                "Id": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
                "AvailabilityZone": "all",
            },
            "TargetHealth": {
                "State": "unavailable",
                "Reason": "Target.HealthCheckDisabled",
                "Description": "Health checks are not enabled for this target"
```

```
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTargetHealthà la section Référence des AWS CLI commandes.

`modify-listener`

L'exemple de code suivant montre comment utiliser`modify-listener`.

AWS CLI

Exemple 1 : pour remplacer l'action par défaut par une action directe

L'`modify-listener`exemple suivant remplace l'action par défaut par une `forward` action pour l'écouteur spécifié.

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```

Sortie :

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTP",
            "Port": 80,
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f"
```

```
                }
            ]
        }
    ]
}
```

Exemple 2 : pour remplacer l'action par défaut par une action de redirection

L'`modify-listener`exemple suivant remplace l'action par défaut par une `redirect` action pour l'écouteur spécifié.

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --default-actions Type=redirect,
 RedirectConfig='{Protocol=HTTPS,StatusCode=HTTP_302}'
```

Sortie :

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTP",
            "Port": 80,
            "DefaultActions": [
                {
                    "Type": "redirect",
                    "RedirectConfig": {
                        "Protocol": "HTTPS",
                        "Port": "#{port}",
                        "Host": "#{host}",
                        "Path": "/#{path}",
                        "Query": "#{query}",
                        "StatusCode": "HTTP_302",
                    }
                }
            ]
```

```
        }
    ]
}
```

Exemple 3 : pour modifier le certificat du serveur

L'`modify-listener`exemple suivant modifie le certificat de serveur pour l'écouteur HTTPS spécifié.

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65 \
    --certificates CertificateArn=arn:aws:iam::123456789012:server-certificate/my-new-server-cert
```

Sortie :

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTPS",
            "Port": 443,
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
                }
            ],
            "SslPolicy": "ELBSecurityPolicy-2015-05",
            "Certificates": [
                {
                    "CertificateArn": "arn:aws:iam::123456789012:server-certificate/my-new-server-cert"
                }
            ],
        }
    ]
```

```
}
```

Pour plus d'informations, consultez les règles relatives aux écouteurs dans le guide de l'utilisateur des équilibreurs de charge d'application.

- Pour plus de détails sur l'API, reportez-vous ModifyListenerà la section Référence des AWS CLI commandes.

`modify-load-balancer-attributes`

L'exemple de code suivant montre comment utiliser`modify-load-balancer-attributes`.

AWS CLI

Pour activer la protection contre les suppressions

Cet exemple active la protection contre la suppression pour l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --attributes Key=deletion_protection.enabled,Value=true
```

Sortie :

```
{
  "Attributes": [
      {
          "Value": "true",
          "Key": "deletion_protection.enabled"
      },
      {
          "Value": "false",
          "Key": "access_logs.s3.enabled"
      },
      {
          "Value": "60",
          "Key": "idle_timeout.timeout_seconds"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.prefix"
```

```
        },
        {
            "Value": "",
            "Key": "access_logs.s3.bucket"
        }
    ]
}
```

Pour modifier le délai d'inactivité

Cet exemple modifie la valeur du délai d'inactivité pour l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --attributes Key=idle_timeout.timeout_seconds,Value=30
```

Sortie :

```
{
  "Attributes": [
      {
          "Value": "30",
          "Key": "idle_timeout.timeout_seconds"
      },
      {
          "Value": "false",
          "Key": "access_logs.s3.enabled"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.prefix"
      },
      {
          "Value": "true",
          "Key": "deletion_protection.enabled"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.bucket"
      }
  ]
```

```
}
```

Pour activer les journaux d'accès

Cet exemple active les journaux d'accès pour l'équilibreur de charge spécifié. Notez que le bucket S3 doit se trouver dans la même région que l'équilibreur de charge et doit être associé à une politique autorisant l'accès au service Elastic Load Balancing.

Commande :

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --
attributes Key=access_logs.s3.enabled,Value=true Key=access_logs.s3.bucket,Value=my-
loadbalancer-logs Key=access_logs.s3.prefix,Value=myapp
```

Sortie :

```
{
  "Attributes": [
      {
          "Value": "true",
          "Key": "access_logs.s3.enabled"
      },
      {
          "Value": "my-load-balancer-logs",
          "Key": "access_logs.s3.bucket"
      },
      {
          "Value": "myapp",
          "Key": "access_logs.s3.prefix"
      },
      {
          "Value": "60",
          "Key": "idle_timeout.timeout_seconds"
      },
      {
          "Value": "false",
          "Key": "deletion_protection.enabled"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyLoadBalancerAttributesà la section Référence des AWS CLI commandes.

`modify-rule`

L'exemple de code suivant montre comment utiliser`modify-rule`.

AWS CLI

Pour modifier une règle

L'`modify-rule`exemple suivant met à jour les actions et les conditions pour la règle spécifiée.

```
aws elbv2 modify-rule \
  --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
  --conditions Field=path-pattern,Values='/images/*'
  --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

Sortie :

```
{
    "Rules": [
        {
            "Priority": "10",
            "Conditions": [
                {
                    "Field": "path-pattern",
                    "Values": [
                        "/images/*"
                    ]
                }
            ],
            "RuleArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2/9683b2d02a6cabee",
            "IsDefault": false,
            "Actions": [
                {
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                    "Type": "forward"
```

```
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyRuleà la section Référence des AWS CLI commandes.

`modify-target-group-attributes`

L'exemple de code suivant montre comment utiliser`modify-target-group-attributes`.

AWS CLI

Pour modifier le délai de désenregistrement

Cet exemple définit le délai de désenregistrement à la valeur spécifiée pour le groupe cible spécifié.

Commande :

```
aws elbv2 modify-target-group-attributes --target-group-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067 --
attributes Key=deregistration_delay.timeout_seconds,Value=600
```

Sortie :

```
{
  "Attributes": [
      {
          "Value": "false",
          "Key": "stickiness.enabled"
      },
      {
          "Value": "600",
          "Key": "deregistration_delay.timeout_seconds"
      },
      {
          "Value": "lb_cookie",
          "Key": "stickiness.type"
```

```
        },
        {
            "Value": "86400",
            "Key": "stickiness.lb_cookie.duration_seconds"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyTargetGroupAttributesà la section Référence des AWS CLI commandes.

`modify-target-group`

L'exemple de code suivant montre comment utiliser`modify-target-group`.

AWS CLI

Pour modifier la configuration du contrôle de santé d'un groupe cible

L'`modify-target-group`exemple suivant modifie la configuration des contrôles de santé utilisés pour évaluer l'état des cibles pour le groupe cible spécifié. Notez qu'en raison de la façon dont la CLI analyse les virgules, vous devez entourer la plage de l'`--matcher`option de guillemets simples plutôt que de guillemets doubles.

```
aws elbv2 modify-target-group \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f \
    --health-check-protocol HTTPS \
    --health-check-port 443 \
    --matcher HttpCode='200,299'
```

Sortie :

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f",
            "TargetGroupName": "my-https-targets",
            "Protocol": "HTTPS",
            "Port": 443,
            "VpcId": "vpc-3ac0fb5f",
```

```
            "HealthCheckProtocol": "HTTPS",
            "HealthCheckPort": "443",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "Matcher": {
                "HttpCode": "200,299"
            },
            "LoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/my-load-balancer/50dc6c495c0c9188"
            ],
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes cibles dans le Guide des équilibreurs de charge des applications.

- Pour plus de détails sur l'API, reportez-vous ModifyTargetGroupà la section Référence des AWS CLI commandes.

`register-targets`

L'exemple de code suivant montre comment utiliser`register-targets`.

AWS CLI

Exemple 1 : pour enregistrer des cibles auprès d'un groupe cible par ID d'instance

L'`register-targets`exemple suivant enregistre les instances spécifiées auprès d'un groupe cible. Le groupe cible doit avoir un type de cible de`instance`.

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

Exemple 2 : pour enregistrer des cibles auprès d'un groupe cible à l'aide de remplacements de port

L'`register-targets`exemple suivant enregistre l'instance spécifiée auprès d'un groupe cible à l'aide de plusieurs ports. Cela vous permet d'enregistrer des conteneurs sur la même instance en tant que cibles du groupe cible.

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \
    --targets Id=i-0598c7d356eba48d7,Port=80 Id=i-0598c7d356eba48d7,Port=766
```

Exemple 3 : pour enregistrer des cibles auprès d'un groupe cible par adresse IP

L'`register-targets`exemple suivant enregistre les adresses IP spécifiées auprès d'un groupe cible. Le groupe cible doit avoir un type de cible de`ip`.

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \
    --targets Id=10.0.1.15 Id=10.0.1.23
```

Exemple 4 : Pour enregistrer une fonction Lambda en tant que cible

L'`register-targets`exemple suivant enregistre les adresses IP spécifiées auprès d'un groupe cible. Le groupe cible doit avoir un type de cible de`lambda`. Vous devez autoriser Elastic Load Balancing à appeler la fonction Lambda.

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \
    --targets Id=arn:aws:lambda:us-west-2:123456789012:function:my-function
```

- Pour plus de détails sur l'API, reportez-vous RegisterTargetsà la section Référence des AWS CLI commandes.

`remove-listener-certificates`

L'exemple de code suivant montre comment utiliser`remove-listener-certificates`.

AWS CLI

Pour supprimer un certificat d'un écouteur sécurisé

Cet exemple supprime le certificat spécifié de l'écouteur sécurisé spécifié.

Commande :

```
aws elbv2 remove-listener-certificates --listener-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --
certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-
f4a3-4072-80be-05b9ba72f705
```

- Pour plus de détails sur l'API, reportez-vous RemoveListenerCertificatesà la section Référence des AWS CLI commandes.

`remove-tags`

L'exemple de code suivant montre comment utiliser`remove-tags`.

AWS CLI

Pour supprimer des balises d'un équilibreur de charge

L'`remove-tags`exemple suivant supprime les `department` balises `project` et de l'équilibreur de charge spécifié.

```
aws elbv2 remove-tags \
    --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --tag-keys project department
```

- Pour plus de détails sur l'API, reportez-vous RemoveTagsà la section Référence des AWS CLI commandes.

`set-ip-address-type`

L'exemple de code suivant montre comment utiliser`set-ip-address-type`.

AWS CLI

Pour définir le type d'adresse d'un équilibreur de charge

Cet exemple définit le type d'adresse de l'équilibreur de charge spécifié sur`dualstack`. Les sous-réseaux de l'équilibreur de charge doivent être associés à des blocs IPv6 CIDR.

Commande :

```
aws elbv2 set-ip-address-type --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --ip-address-type dualstack
```

Sortie :

```
{
    "IpAddressType": "dualstack"
}
```

- Pour plus de détails sur l'API, reportez-vous SetIpAddressTypeà la section Référence des AWS CLI commandes.

`set-rule-priorities`

L'exemple de code suivant montre comment utiliser`set-rule-priorities`.

AWS CLI

Pour définir la priorité de la règle

Cet exemple définit la priorité de la règle spécifiée.

Commande :

```
aws elbv2 set-rule-priorities --rule-priorities RuleArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3,Priority=5
```

Sortie :

```
{
```

```
  "Rules": [
      {
          "Priority": "5",
          "Conditions": [
              {
                  "Field": "path-pattern",
                  "Values": [
                      "/img/*"
                  ]
              }
          ],
          "RuleArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-
rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3",
          "IsDefault": false,
          "Actions": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                  "Type": "forward"
              }
          ]
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous SetRulePrioritiesà la section Référence des AWS CLI commandes.

`set-security-groups`

L'exemple de code suivant montre comment utiliser`set-security-groups`.

AWS CLI

Pour associer un groupe de sécurité à un équilibreur de charge

Cet exemple associe le groupe de sécurité spécifié à l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 set-security-groups --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --security-
groups sg-5943793c
```

Sortie :

```
{
  "SecurityGroupIds": [
      "sg-5943793c"
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous SetSecurityGroupsà la section Référence des AWS CLI commandes.

`set-subnets`

L'exemple de code suivant montre comment utiliser`set-subnets`.

AWS CLI

Pour activer les zones de disponibilité pour un équilibreur de charge

Cet exemple active la zone de disponibilité pour le sous-réseau spécifié pour l'équilibreur de charge spécifié.

Commande :

```
aws elbv2 set-subnets --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --
subnets subnet-8360a9e7 subnet-b7d581c0
```

Sortie :

```
{
  "AvailabilityZones": [
      {
          "SubnetId": "subnet-8360a9e7",
          "ZoneName": "us-west-2a"
      },
      {
          "SubnetId": "subnet-b7d581c0",
          "ZoneName": "us-west-2b"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous SetSubnetsà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Elastic Transcoder AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Elastic Transcoder.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-job`

L'exemple de code suivant montre comment utiliser`cancel-job`.

AWS CLI

Pour annuler une offre d'emploi pour ElasticTranscoder

Cela annule la tâche spécifiée pour ElasticTranscoder.

Commande :

```
aws elastictranscoder cancel-job --id 3333333333333-abcde3
```

- Pour plus de détails sur l'API, voir CancelJobla section Référence des AWS CLI commandes.

`create-job`

L'exemple de code suivant montre comment utiliser`create-job`.

AWS CLI

Pour créer un emploi pour ElasticTranscoder

L'`create-job`exemple suivant crée une tâche pour ElasticTranscoder.

```
aws elastictranscoder create-job \
    --pipeline-id 1111111111111-abcde1 \
    --inputs file://inputs.json \
    --outputs file://outputs.json \
    --output-key-prefix "recipes/" \
    --user-metadata file://user-metadata.json
```

Contenu de `inputs.json` :

```
[{
    "Key":"ETS_example_file.mp4",
    "FrameRate":"auto",
    "Resolution":"auto",
    "AspectRatio":"auto",
    "Interlaced":"auto",
    "Container":"mp4"
}]
```

Contenu de outputs.json :

```
[
    {
        "Key":"webm/ETS_example_file-kindlefirehd.webm",
        "Rotate":"0",
        "PresetId":"1351620000001-100250"
    }
]
```

Contenu de `user-metadata.json` :

```
{
    "Food type":"Italian",
    "Cook book":"recipe notebook"
}
```

Sortie :


```
{
    "Job": {
        "Status": "Submitted",
        "Inputs": [
            {
                "Container": "mp4",
                "FrameRate": "auto",
                "Key": "ETS_example_file.mp4",
                "AspectRatio": "auto",
                "Resolution": "auto",
                "Interlaced": "auto"
            }
        ],
        "Playlists": [],
        "Outputs": [
            {
                "Status": "Submitted",
                "Rotate": "0",
                "PresetId": "1351620000001-100250",
                "Watermarks": [],
                "Key": "webm/ETS_example_file-kindlefirehd.webm",
                "Id": "1"
            }
        ],
        "PipelineId": "3333333333333-abcde3",
        "OutputKeyPrefix": "recipes/",
        "UserMetadata": {
            "Cook book": "recipe notebook",
            "Food type": "Italian"
        },
        "Output": {
            "Status": "Submitted",
            "Rotate": "0",
            "PresetId": "1351620000001-100250",
            "Watermarks": [],
            "Key": "webm/ETS_example_file-kindlefirehd.webm",
            "Id": "1"
        },
        "Timing": {
            "SubmitTimeMillis": 1533838012298
        },
        "Input": {
```

```
            "Container": "mp4",
            "FrameRate": "auto",
            "Key": "ETS_example_file.mp4",
            "AspectRatio": "auto",
            "Resolution": "auto",
            "Interlaced": "auto"
        },
        "Id": "1533838012294-example",
        "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}
```

- Pour plus de détails sur l'API, voir CreateJobla section Référence des AWS CLI commandes.

`create-pipeline`

L'exemple de code suivant montre comment utiliser`create-pipeline`.

AWS CLI

Pour créer un pipeline pour ElasticTranscoder

L'`create-pipeline`exemple suivant crée un pipeline pour ElasticTranscoder.

```
aws elastictranscoder create-pipeline \
    --name Default \
    --input-bucket salesoffice.example.com-source \
    --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
    --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
    --content-config file://content-config.json \
    --thumbnail-config file://thumbnail-config.json
```

Contenu de `content-config.json` :

```
{
    "Bucket":"salesoffice.example.com-public-promos",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
```

```
            "Access":[
            "FullControl"
            ]
        }
    ],
    "StorageClass":"Standard"
}
```

Contenu de `thumbnail-config.json` :

```
{
    "Bucket":"salesoffice.example.com-public-promos-thumbnails",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"ReducedRedundancy"
}
```

Sortie :

```
{
    "Pipeline": {
        "Status": "Active",
        "ContentConfig": {
            "Bucket": "salesoffice.example.com-public-promos",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "Default",
```

```
        "ThumbnailConfig": {
            "Bucket": "salesoffice.example.com-public-promos-thumbnails",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "salesoffice.example.com-source",
        "Id": "1533765810590-example",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1533765810590-example"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir CreatePipelinela section Référence des AWS CLI commandes.

`create-preset`

L'exemple de code suivant montre comment utiliser`create-preset`.

AWS CLI

Pour créer un préréglage pour ElasticTranscoder

L'`create-preset`exemple suivant crée un préréglage pour ElasticTranscoder.

```
aws elastictranscoder create-preset \
    --name DefaultPreset \
    --description "Use for published videos" \
    --container mp4 \
    --video file://video.json \
    --audio file://audio.json \
    --thumbnails file://thumbnails.json
```

Contenu de `video.json` :

```
{
    "Codec":"H.264",
    "CodecOptions":{
        "Profile":"main",
        "Level":"2.2",
        "MaxReferenceFrames":"3",
        "MaxBitRate":"",
        "BufferSize":"",
        "InterlacedMode":"Progressive",
        "ColorSpaceConversionMode":"None"
    },
    "KeyframesMaxDist":"240",
    "FixedGOP":"false",
    "BitRate":"1600",
    "FrameRate":"auto",
    "MaxFrameRate":"30",
    "MaxWidth":"auto",
    "MaxHeight":"auto",
    "SizingPolicy":"Fit",
    "PaddingPolicy":"Pad",
    "DisplayAspectRatio":"auto",
    "Watermarks":[
        {
            "Id":"company logo",
            "MaxWidth":"20%",
            "MaxHeight":"20%",
            "SizingPolicy":"ShrinkToFit",
```

```
            "HorizontalAlign":"Right",
            "HorizontalOffset":"10px",
            "VerticalAlign":"Bottom",
            "VerticalOffset":"10px",
            "Opacity":"55.5",
            "Target":"Content"
        }
    ]
}
```

Contenu de `audio.json` :

```
{
    "Codec":"AAC",
    "CodecOptions":{
        "Profile":"AAC-LC"
    },
    "SampleRate":"44100",
    "BitRate":"96",
    "Channels":"2"
}
```

Contenu de `thumbnails.json` :

```
{
    "Format":"png",
    "Interval":"120",
    "MaxWidth":"auto",
    "MaxHeight":"auto",
    "SizingPolicy":"Fit",
    "PaddingPolicy":"Pad"
}
```

Sortie :

```
{
    "Preset": {
        "Thumbnails": {
            "SizingPolicy": "Fit",
            "MaxWidth": "auto",
            "Format": "png",
            "PaddingPolicy": "Pad",
```

```
            "Interval": "120",
            "MaxHeight": "auto"
        },
        "Container": "mp4",
        "Description": "Use for published videos",
        "Video": {
            "SizingPolicy": "Fit",
            "MaxWidth": "auto",
            "PaddingPolicy": "Pad",
            "MaxFrameRate": "30",
            "FrameRate": "auto",
            "MaxHeight": "auto",
            "KeyframesMaxDist": "240",
            "FixedGOP": "false",
            "Codec": "H.264",
            "Watermarks": [
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10px",
                    "VerticalAlign": "Bottom",
                    "Target": "Content",
                    "MaxWidth": "20%",
                    "MaxHeight": "20%",
                    "HorizontalAlign": "Right",
                    "HorizontalOffset": "10px",
                    "Opacity": "55.5",
                    "Id": "company logo"
                }
            ],
            "CodecOptions": {
                "Profile": "main",
                "MaxBitRate": "32",
                "InterlacedMode": "Progressive",
                "Level": "2.2",
                "ColorSpaceConversionMode": "None",
                "MaxReferenceFrames": "3",
                "BufferSize": "5"
            },
            "BitRate": "1600",
            "DisplayAspectRatio": "auto"
        },
        "Audio": {
            "Channels": "2",
            "CodecOptions": {
```

```
                "Profile": "AAC-LC"
            },
            "SampleRate": "44100",
            "Codec": "AAC",
            "BitRate": "96"
        },
        "Type": "Custom",
        "Id": "1533765290724-example"
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1533765290724-example",
        "Name": "DefaultPreset"
    },
    "Warning": ""
}
```

- Pour plus de détails sur l'API, voir CreatePresetla section Référence des AWS CLI commandes.

`delete-pipeline`

L'exemple de code suivant montre comment utiliser`delete-pipeline`.

AWS CLI

Pour supprimer le ElasticTranscoder pipeline spécifié

Cela supprime le ElasticTranscoder pipeline spécifié.

Commande :

```
aws elastictranscoder delete-pipeline --id 1111111111111-abcde1
```

Sortie :

```
{
 "Success":"true"
}
```

- Pour plus de détails sur l'API, voir DeletePipelinela section Référence des AWS CLI commandes.

`delete-preset`

L'exemple de code suivant montre comment utiliser`delete-preset`.

AWS CLI

Pour supprimer le ElasticTranscoder préréglage spécifié

Cela supprime le ElasticTranscoder préréglage spécifié.

Commande :

```
aws elastictranscoder delete-preset --id 5555555555555-abcde5
```

- Pour plus de détails sur l'API, voir DeletePresetla section Référence des AWS CLI commandes.

`list-jobs-by-pipeline`

L'exemple de code suivant montre comment utiliser`list-jobs-by-pipeline`.

AWS CLI

Pour récupérer une liste de ElasticTranscoder tâches dans le pipeline spécifié

Cet exemple extrait une liste de ElasticTranscoder tâches dans le pipeline spécifié.

Commande :

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 1111111111111-abcde1
```

Sortie :

```
{
   "Jobs": []
}
```

- Pour plus de détails sur l'API, voir ListJobsByPipelinela section Référence des AWS CLI commandes.

`list-jobs-by-status`

L'exemple de code suivant montre comment utiliser`list-jobs-by-status`.

AWS CLI

Pour récupérer une liste de ElasticTranscoder tâches dont le statut est Terminé

Cet exemple extrait une liste de ElasticTranscoder tâches dont le statut est Terminé.

Commande :

```
aws elastictranscoder list-jobs-by-status --status Complete
```

Sortie :

```
{
   "Jobs": []
}
```

- Pour plus de détails sur l'API, voir ListJobsByStatusla section Référence des AWS CLI commandes.

`list-pipelines`

L'exemple de code suivant montre comment utiliser`list-pipelines`.

AWS CLI

Pour récupérer une liste de ElasticTranscoder pipelines

Cet exemple extrait une liste de ElasticTranscoder pipelines.

Commande :

```
aws elastictranscoder list-pipelines
```

Sortie :

```
{
   "Pipelines": [
       {
           "Status": "Active",
           "ContentConfig": {
               "Bucket": "ets-example",
               "Permissions": []
```

```
            },
            "Name": "example-pipeline",
            "ThumbnailConfig": {
                "Bucket": "ets-example",
                "Permissions": []
            },
            "Notifications": {
                "Completed": "arn:aws:sns:us-west-2:123456789012:ets_example",
                "Warning": "",
                "Progressing": "",
                "Error": ""
            },
            "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
            "InputBucket": "ets-example",
            "OutputBucket": "ets-example",
            "Id": "3333333333333-abcde3",
            "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
        },
        {
            "Status": "Paused",
            "ContentConfig": {
                "Bucket": "ets-example",
                "Permissions": []
            },
            "Name": "example-php-test",
            "ThumbnailConfig": {
                "Bucket": "ets-example",
                "Permissions": []
            },
            "Notifications": {
                "Completed": "",
                "Warning": "",
                "Progressing": "",
                "Error": ""
            },
            "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
            "InputBucket": "ets-example",
            "OutputBucket": "ets-example",
            "Id": "3333333333333-abcde2",
            "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde2"
        },
        {
```

```
            "Status": "Active",
            "ContentConfig": {
                "Bucket": "ets-west-output",
                "Permissions": []
            },
            "Name": "pipeline-west",
            "ThumbnailConfig": {
                "Bucket": "ets-west-output",
                "Permissions": []
            },
            "Notifications": {
                "Completed": "arn:aws:sns:us-west-2:123456789012:ets-notifications",
                "Warning": "",
                "Progressing": "",
                "Error": ""
            },
            "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
            "InputBucket": "ets-west-input",
            "OutputBucket": "ets-west-output",
            "Id": "3333333333333-abcde1",
            "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde1"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListPipelinesla section Référence des AWS CLI commandes.

`list-presets`

L'exemple de code suivant montre comment utiliser`list-presets`.

AWS CLI

Pour récupérer une liste de ElasticTranscoder préréglages

Cet exemple permet de récupérer une liste de ElasticTranscoder préréglages.

Commande :

```
aws elastictranscoder list-presets --max-items 2
```

Sortie :

```
{
  "Presets": [
      {
          "Container": "mp4",
          "Name": "KindleFireHD-preset",
          "Video": {
              "Resolution": "1280x720",
              "FrameRate": "30",
              "KeyframesMaxDist": "90",
              "FixedGOP": "false",
              "Codec": "H.264",
              "Watermarks": [],
              "CodecOptions": {
                  "Profile": "main",
                  "MaxReferenceFrames": "3",
                  "ColorSpaceConversionMode": "None",
                  "InterlacedMode": "Progressive",
                  "Level": "4"
              },
              "AspectRatio": "16:9",
              "BitRate": "2200"
          },
          "Audio": {
              "Channels": "2",
              "CodecOptions": {
                  "Profile": "AAC-LC"
              },
              "SampleRate": "48000",
              "Codec": "AAC",
              "BitRate": "160"
          },
          "Type": "Custom",
          "Id": "3333333333333-abcde2",
          "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/3333333333333-abcde2",
          "Thumbnails": {
              "AspectRatio": "16:9",
              "Interval": "60",
              "Resolution": "192x108",
              "Format": "png"
          }
      },
      {
```

```
          "Thumbnails": {
              "AspectRatio": "16:9",
              "Interval": "60",
              "Resolution": "192x108",
              "Format": "png"
          },
          "Container": "mp4",
          "Description": "Custom preset for transcoding jobs",
          "Video": {
              "Resolution": "1280x720",
              "FrameRate": "30",
              "KeyframesMaxDist": "90",
              "FixedGOP": "false",
              "Codec": "H.264",
              "Watermarks": [],
              "CodecOptions": {
                  "Profile": "main",
                  "MaxReferenceFrames": "3",
                  "ColorSpaceConversionMode": "None",
                  "InterlacedMode": "Progressive",
                  "Level": "3.1"
              },
              "AspectRatio": "16:9",
              "BitRate": "2200"
          },
          "Audio": {
              "Channels": "2",
              "CodecOptions": {
                  "Profile": "AAC-LC"
              },
              "SampleRate": "44100",
              "Codec": "AAC",
              "BitRate": "160"
          },
          "Type": "Custom",
          "Id": "3333333333333-abcde3",
          "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/3333333333333-abcde3",
          "Name": "Roman's Preset"
      }
  ],
  "NextToken": "eyJQYWdlVG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- Pour plus de détails sur l'API, voir ListPresetsla section Référence des AWS CLI commandes.

`read-job`

L'exemple de code suivant montre comment utiliser`read-job`.

AWS CLI

Pour récupérer une ElasticTranscoder tâche

Cet exemple récupère la ElasticTranscoder tâche spécifiée.

Commande :

```
aws elastictranscoder read-job --id 1533838012294-example
```

Sortie :

```
{
   "Job": {
       "Status": "Progressing",
       "Inputs": [
           {
               "Container": "mp4",
               "FrameRate": "auto",
               "Key": "ETS_example_file.mp4",
               "AspectRatio": "auto",
               "Resolution": "auto",
               "Interlaced": "auto"
           }
       ],
       "Playlists": [],
       "Outputs": [
           {
               "Status": "Progressing",
               "Rotate": "0",
               "PresetId": "1351620000001-100250",
               "Watermarks": [],
               "Key": "webm/ETS_example_file-kindlefirehd.webm",
               "Id": "1"
           }
       ],
```

```
        "PipelineId": "3333333333333-abcde3",
        "OutputKeyPrefix": "recipes/",
        "UserMetadata": {
            "Cook book": "recipe notebook",
            "Food type": "Italian"
        },
        "Output": {
            "Status": "Progressing",
            "Rotate": "0",
            "PresetId": "1351620000001-100250",
            "Watermarks": [],
            "Key": "webm/ETS_example_file-kindlefirehd.webm",
            "Id": "1"
        },
        "Timing": {
            "SubmitTimeMillis": 1533838012298,
            "StartTimeMillis": 1533838013786
        },
        "Input": {
            "Container": "mp4",
            "FrameRate": "auto",
            "Key": "ETS_example_file.mp4",
            "AspectRatio": "auto",
            "Resolution": "auto",
            "Interlaced": "auto"
        },
        "Id": "1533838012294-example",
        "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}
```

- Pour plus de détails sur l'API, voir ReadJobla section Référence des AWS CLI commandes.

`read-pipeline`

L'exemple de code suivant montre comment utiliser`read-pipeline`.

AWS CLI

Pour récupérer un ElasticTranscoder pipeline

Cet exemple extrait le ElasticTranscoder pipeline spécifié.

Commande :

```
aws elastictranscoder read-pipeline --id 3333333333333-abcde3
```

Sortie :

```
{
   "Pipeline": {
       "Status": "Active",
       "ContentConfig": {
           "Bucket": "ets-example",
           "StorageClass": "Standard",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
                   "Grantee": "marketing-promos@example.com",
                   "GranteeType": "Email"
               }
           ]
       },
       "Name": "Default",
       "ThumbnailConfig": {
           "Bucket": "ets-example",
           "StorageClass": "ReducedRedundancy",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
                   "Grantee": "marketing-promos@example.com",
                   "GranteeType": "Email"
               }
           ]
       },
       "Notifications": {
           "Completed": "",
           "Warning": "",
           "Progressing": "",
           "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
       },
       "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
```

```
        "InputBucket": "ets-example",
        "Id": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ReadPipelinela section Référence des AWS CLI commandes.

`read-preset`

L'exemple de code suivant montre comment utiliser`read-preset`.

AWS CLI

Pour récupérer un ElasticTranscoder préréglage

Cet exemple extrait le ElasticTranscoder préréglage spécifié.

Commande :

```
aws elastictranscoder read-preset --id 1351620000001-500020
```

Sortie :

```
{
    "Preset": {
        "Thumbnails": {
            "SizingPolicy": "ShrinkToFit",
            "MaxWidth": "192",
            "Format": "png",
            "PaddingPolicy": "NoPad",
```

```
            "Interval": "300",
            "MaxHeight": "108"
        },
        "Container": "fmp4",
        "Description": "System preset: MPEG-Dash Video - 4.8M",
        "Video": {
            "SizingPolicy": "ShrinkToFit",
            "MaxWidth": "1280",
            "PaddingPolicy": "NoPad",
            "FrameRate": "30",
            "MaxHeight": "720",
            "KeyframesMaxDist": "60",
            "FixedGOP": "true",
            "Codec": "H.264",
            "Watermarks": [
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Top",
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Left",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "TopLeft"
                },
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Top",
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Right",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "TopRight"
                },
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Bottom",
                    "Target": "Content",
```

```
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Left",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "BottomLeft"
                },
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Bottom",
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Right",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "BottomRight"
                }
            ],
            "CodecOptions": {
                "Profile": "main",
                "MaxBitRate": "4800",
                "InterlacedMode": "Progressive",
                "Level": "3.1",
                "ColorSpaceConversionMode": "None",
                "MaxReferenceFrames": "3",
                "BufferSize": "9600"
            },
            "BitRate": "4800",
            "DisplayAspectRatio": "auto"
        },
        "Type": "System",
        "Id": "1351620000001-500020",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1351620000001-500020",
        "Name": "System preset: MPEG-Dash Video - 4.8M"
    }
}
```

- Pour plus de détails sur l'API, voir ReadPresetla section Référence des AWS CLI commandes.

`update-pipeline-notifications`

L'exemple de code suivant montre comment utiliser`update-pipeline-notifications`.

AWS CLI

Pour mettre à jour les notifications d'un ElasticTranscoder pipeline

Cet exemple met à jour les notifications du ElasticTranscoder pipeline spécifié.

Commande :

```
aws elastictranscoder update-pipeline-notifications --id 1111111111111-abcde1 --notifications Progressing=arn:aws:sns:us-west-2:0123456789012:my-topic,Completed=arn:aws:sns:us-west-2:0123456789012:my-topic,Warning=arn:aws:sns:us-west-2:0123456789012:my-topic,Error=arn:aws:sns:us-east-1:111222333444:ETS_Errors
```

Sortie :

```
{
   "Pipeline": {
       "Status": "Active",
       "ContentConfig": {
           "Bucket": "ets-example",
           "StorageClass": "Standard",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
                   "Grantee": "marketing-promos@example.com",
                   "GranteeType": "Email"
               }
           ]
       },
       "Name": "Default",
       "ThumbnailConfig": {
           "Bucket": "ets-example",
           "StorageClass": "ReducedRedundancy",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
```

```
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "arn:aws:sns:us-west-2:0123456789012:my-topic",
            "Warning": "arn:aws:sns:us-west-2:0123456789012:my-topic",
            "Progressing": "arn:aws:sns:us-west-2:0123456789012:my-topic",
            "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "1111111111111-abcde1",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1111111111111-abcde1"
    }
}
```

- Pour plus de détails sur l'API, voir UpdatePipelineNotificationsla section Référence des AWS CLI commandes.

`update-pipeline-status`

L'exemple de code suivant montre comment utiliser`update-pipeline-status`.

AWS CLI

Pour mettre à jour le statut d'un ElasticTranscoder pipeline

Cet exemple met à jour l'état du ElasticTranscoder pipeline spécifié.

Commande :

```
aws elastictranscoder update-pipeline-status --id 1111111111111-abcde1 --
status Paused
```

Sortie :

```
{
    "Pipeline": {
        "Status": "Paused",
```

```
        "ContentConfig": {
            "Bucket": "ets-example",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "Default",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:803981987763:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "1111111111111-abcde1",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1111111111111-abcde1"
    }
}
```

- Pour plus de détails sur l'API, voir UpdatePipelineStatusla section Référence des AWS CLI commandes.

update-pipeline

L'exemple de code suivant montre comment utiliserupdate-pipeline.

AWS CLI

Pour mettre à jour un ElasticTranscoder pipeline

L'update-pipelineexemple suivant met à jour le ElasticTranscoder pipeline spécifié.

```
aws elastictranscoder update-pipeline \
    --id 1111111111111-abcde1
    --name DefaultExample \
    --input-bucket salesoffice.example.com-source \
    --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
    --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
    --content-config file://content-config.json \
    --thumbnail-config file://thumbnail-config.json
```

Contenu de content-config.json :

```
{
    "Bucket":"salesoffice.example.com-public-promos",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"Standard"
}
```

Contenu de thumbnail-config.json :

```
{
    "Bucket":"salesoffice.example.com-public-promos-thumbnails",
    "Permissions":[
        {
```

```
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"ReducedRedundancy"
}
```

Sortie :

```
{
    "Pipeline": {
        "Status": "Active",
        "ContentConfig": {
            "Bucket": "ets-example",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "DefaultExample",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
```

```
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir UpdatePipelinela section Référence des AWS CLI commandes.

ElastiCache exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with ElastiCache.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-resource`

L'exemple de code suivant montre comment utiliser`add-tags-to-resource`.

AWS CLI

Pour ajouter des balises à une ressource

L'`add-tags-to-resource`exemple suivant ajoute jusqu'à 10 balises, paires clé-valeur, à un cluster ou à une ressource de capture instantanée.

```
aws elasticache add-tags-to-resource \
    --resource-name "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-
cluster" \
    --tags '{"20150202":15, "ElastiCache":"Service"}'
```

Sortie :

```
{
    "TagList": [
        {
            "Value": "20150202",
            "Key": "APIVersion"
        },
        {
            "Value": "ElastiCache",
            "Key": "Service"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance des coûts à l'aide de balises de répartition des coûts dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous AddTagsToResourceà la section Référence des AWS CLI commandes.

`authorize-cache-security-group-ingress`

L'exemple de code suivant montre comment utiliser`authorize-cache-security-group-ingress`.

AWS CLI

Pour autoriser l'entrée du groupe de sécurité du cache

L'`authorize-cache-security-group-ingress`exemple suivant autorise l'accès du réseau à un groupe de sécurité du cache.

```
aws elasticache authorize-cache-security-group-ingress \
     --cache-security-group-name  "my-sec-grp" \
     --ec2-security-group-name "my-ec2-sec-grp" \
     --ec2-security-group-owner-id "1234567890"
```

La commande ne produit aucune sortie.

Pour plus d'informations, consultez les mises à jour en libre-service sur Amazon ElastiCache dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous AuthorizeCacheSecurityGroupIngressà la section Référence des AWS CLI commandes.

`batch-apply-update-action`

L'exemple de code suivant montre comment utiliser`batch-apply-update-action`.

AWS CLI

Pour appliquer une mise à jour de service

L'`batch-apply-update-action`exemple suivant applique une mise à jour de service à un cluster Redis.

```
aws elasticache batch-apply-update-action \
    --service-update-name elc-xxxxx406-xxx \
    --replication-group-ids test-cluster
```

Sortie :

```
{
    "ProcessedUpdateActions": [
        {
            "ReplicationGroupId": "pat-cluster",
            "ServiceUpdateName": "elc-xxxxx406-xxx",
            "UpdateActionStatus": "waiting-to-start"
        }
    ],
    "UnprocessedUpdateActions": []
}
```

Pour plus d'informations, consultez les mises à jour en libre-service sur Amazon ElastiCache dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous BatchApplyUpdateActionà la section Référence des AWS CLI commandes.

`batch-stop-update-action`

L'exemple de code suivant montre comment utiliser`batch-stop-update-action`.

AWS CLI

Pour arrêter une mise à jour de service

L'`batch-stop-update-action`exemple suivant applique une mise à jour de service à un cluster Redis.

```
aws elasticache batch-stop-update-action \
    --service-update-name elc-xxxxx406-xxx \
    --replication-group-ids test-cluster
```

Sortie :

```
{
    "ProcessedUpdateActions": [
        {
            "ReplicationGroupId": "pat-cluster",
            "ServiceUpdateName": "elc-xxxxx406-xxx",
            "UpdateActionStatus": "stopping"
        }
    ],
```

```
    "UnprocessedUpdateActions": []
}
```

Pour plus d'informations, consultez les mises à jour en libre-service sur Amazon ElastiCache dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous BatchStopUpdateActionà la section Référence des AWS CLI commandes.

`copy-snapshot`

L'exemple de code suivant montre comment utiliser`copy-snapshot`.

AWS CLI

Pour copier un instantané

L'`copy-snapshot`exemple suivant crée une copie d'un instantané existant.

```
aws elasticache copy-snapshot \
    --source-snapshot-name "my-snapshot" \
    --target-snapshot-name "my-snapshot-copy"
```

Sortie :

```
{
    "Snapshot":{
        "Engine": "redis",
        "CacheParameterGroupName": "default.redis3.2",
        "VpcId": "vpc-3820329f3",
        "CacheClusterId": "my-redis4",
        "SnapshotRetentionLimit": 7,
        "NumCacheNodes": 1,
        "SnapshotName": "my-snapshot-copy",
        "CacheClusterCreateTime": "2016-12-21T22:24:04.955Z",
        "AutoMinorVersionUpgrade": true,
        "PreferredAvailabilityZone": "us-east-1c",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "SnapshotWindow": "07:00-08:00",
        "EngineVersion": "3.2.4",
        "NodeSnapshots": [
            {
```

```
                "CacheSize": "3 MB",
                "SnapshotCreateTime": "2016-12-28T07:00:52Z",
                "CacheNodeId": "0001",
                "CacheNodeCreateTime": "2016-12-21T22:24:04.955Z"
            }
        ],
        "CacheSubnetGroupName": "default",
        "Port": 6379,
        "PreferredMaintenanceWindow": "tue:09:30-tue:10:30",
        "CacheNodeType": "cache.m3.large"
    }
}
```

Pour plus d'informations, consultez la section Exportation d'une sauvegarde dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CopySnapshotà la section Référence des AWS CLI commandes.

`create-cache-cluster`

L'exemple de code suivant montre comment utiliser`create-cache-cluster`.

AWS CLI

Pour créer un cluster de cache

L'`create-cache-cluster`exemple suivant crée un cluster de cache à l'aide du moteur Redis.

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine redis \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1
```

Sortie :

```
{
    "CacheCluster": {
        "CacheClusterId": "cluster-test",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.m5.large",
```

```
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "creating",
        "NumCacheNodes": 1,
        "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",
        "PendingModifiedValues": {},
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "default.redis5.0",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "default",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:30-07:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Création d'un cluster dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateCacheClusterà la section Référence des AWS CLI commandes.

`create-cache-parameter-group`

L'exemple de code suivant montre comment utiliser`create-cache-parameter-group`.

AWS CLI

Pour créer un groupe de paramètres de cache

L'`create-cache-parameter-group`exemple suivant crée un nouveau groupe de paramètres de ElastiCache cache Amazon.

```
aws elasticache create-cache-parameter-group \
    --cache-parameter-group-family "redis5.0" \
    --cache-parameter-group-name "mygroup" \
    --description "mygroup"
```

Sortie :

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "mygroup",
        "CacheParameterGroupFamily": "redis5.0",
        "Description": "my group"
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de paramètres dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateCacheParameterGroupà la section Référence des AWS CLI commandes.

`create-cache-subnet-group`

L'exemple de code suivant montre comment utiliser`create-cache-subnet-group`.

AWS CLI

Pour créer un groupe de sous-réseaux de cache

L'`create-cache-subnet-group`exemple suivant crée un nouveau groupe de sous-réseaux de cache.

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name "mygroup" \
    --cache-subnet-group-description "my subnet group" \
    --subnet-ids "subnet-xxxxec4f"
```

Sortie :

```
{
    "CacheSubnetGroup": {
        "CacheSubnetGroupName": "mygroup",
        "CacheSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-a3e97cdb",
        "Subnets": [
            {
```

```
                "SubnetIdentifier": "subnet-xxxxec4f",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de sous-réseaux de cache dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateCacheSubnetGroupà la section Référence des AWS CLI commandes.

`create-global-replication-group`

L'exemple de code suivant montre comment utiliser`create-global-replication-group`.

AWS CLI

Pour créer un groupe de réplication global

L'`create-global-replication-group`exemple suivant crée un nouveau groupe de réplication global.

```
aws elasticache create-global-replication-group \
    --global-replication-group-id-suffix my-global-replication-group \
    --primary-replication-group-id my-primary-cluster
```

Sortie :

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-my-global-replication-group",
        "GlobalReplicationGroupDescription": " ",
        "Status": "creating",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
```

```
                "ReplicationGroupId": "my-primary-cluster",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associating"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-my-global-replication-group-0001",
                "Slots": "0-16383"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateGlobalReplicationGroupà la section Référence des AWS CLI commandes.

`create-replication-group`

L'exemple de code suivant montre comment utiliser`create-replication-group`.

AWS CLI

Pour créer un groupe de réplication

L'`create-replication-group`exemple suivant crée un groupe de réplication Redis (mode cluster désactivé) ou Redis (mode cluster activé). Cette opération n'est valide que pour Redis.

```
aws elasticache create-replication-group \
    --replication-group-id "mygroup" \
    --replication-group-description "my group" \
    --engine "redis" \
    --cache-node-type "cache.m5.large"
```

Sortie :

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "mygroup",
        "Description": "my group",
        "Status": "creating",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "mygroup-001"
        ],
        "AutomaticFailover": "disabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:00-07:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.m5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de réplication Redis dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateReplicationGroupà la section Référence des AWS CLI commandes.

`create-snapshot`

L'exemple de code suivant montre comment utiliser`create-snapshot`.

AWS CLI

Pour créer un instantané

L'`create-snapshot`exemple suivant crée un instantané à l'aide du moteur Redis.

```
aws elasticache create-snapshot \
    --snapshot-name mysnapshot \
    --cache-cluster-id cluster-test
```

Sortie :

```
{
    "Snapshot": {
        "SnapshotName": "mysnapshot",
        "CacheClusterId": "cluster-test",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "CacheNodeType": "cache.m5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheClusterCreateTime": "2020-03-19T03:12:01.483Z",
        "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",
        "Port": 6379,
        "CacheParameterGroupName": "default.redis5.0",
        "CacheSubnetGroupName": "default",
        "VpcId": "vpc-a3e97cdb",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:30-07:30",
        "NodeSnapshots": [
            {
                "CacheNodeId": "0001",
                "CacheSize": "",
                "CacheNodeCreateTime": "2020-03-19T03:12:01.483Z"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Backup and Restore ElastiCache for Redis dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateSnapshotà la section Référence des AWS CLI commandes.

`create-user-group`

L'exemple de code suivant montre comment utiliser`create-user-group`.

AWS CLI

Pour créer un groupe d'utilisateurs

L'`create-user-group`exemple suivant crée un nouveau groupe d'utilisateurs.

```
aws elasticache create-user-group \
    --user-group-id myusergroup \
    --engine redis \
    --user-ids default
```

Sortie :

```
{
    "UserGroupId": "myusergroup",
    "Status": "creating",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateUserGroupà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un utilisateur

L'`create-user`exemple suivant crée un nouvel utilisateur.

```
aws elasticache create-user \
    --user-id user1 \
    --user-name myUser \
    --passwords mYnuUzrpAxXw2rdzx \
    --engine redis \
    --access-string "on ~app::* -@all +@read"
```

Sortie :

```
{
    "UserId": "user2",
    "UserName": "myUser",
    "Status": "active",
    "Engine": "redis",
    "AccessString": "on ~app::* -@all +@read +@hash +@bitmap +@geo -setbit -bitfield
 -hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius -
georadiusbymember",
    "UserGroupIds": [],
    "Authentication": {
        "Type": "password",
        "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`decrease-node-groups-in-global-replication-group`

L'exemple de code suivant montre comment utiliser`decrease-node-groups-in-global-replication-group`.

AWS CLI

Pour réduire le nombre de groupes de nœuds dans un groupe de réplication global

Ce qui suit permet `decrease-node-groups-in-global-replication-group` de réduire le nombre de groupes de nœuds à l'aide du moteur Redis.

```
aws elasticache decrease-node-groups-in-global-replication-group \
    --global-replication-group-id sgaui-test \
    --node-group-count 1 \
    --apply-immediately \
    --global-node-groups-to-retain sgaui-test-0003
```

Sortie :


```
{
    "GlobalReplicationGroup":
    {
        "GlobalReplicationGroupId": "sgaui-test",
        "GlobalReplicationGroupDescription": "test",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "test-2",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "test-1",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-test-0001",
                "Slots": "0-449,1816-5461"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0002",
                "Slots": "6827-10922"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0003",
                "Slots": "10923-14052,15418-16383"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0004",
                "Slots": "450-1815,5462-6826,14053-15417"
```

```
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DecreaseNodeGroupsInGlobalReplicationGroupà la section Référence des AWS CLI commandes.

`decrease-replica-count`

L'exemple de code suivant montre comment utiliser`decrease-replica-count`.

AWS CLI

Pour réduire le nombre de répliques

L'`decrease-replica-count`exemple suivant réduit dynamiquement le nombre de répliques dans un groupe de réplication Redis (mode cluster désactivé) ou le nombre de nœuds de réplication dans un ou plusieurs groupes de nœuds (shards) d'un groupe de réplication Redis (mode cluster activé). Cette opération est effectuée sans interruption du cluster.

```
aws elasticache decrease-replica-count \
    --replication-group-id my-cluster \
    --apply-immediately  \
    --new-replica-count 2
```

Sortie :

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "my-cluster",
        "Description": " ",
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "myrepliace",
```

```
            "my-cluster-001",
            "my-cluster-002",
            "my-cluster-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "modifying",
                "PrimaryEndpoint": {
                    "Address": "my-cluster.xxxxx.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "my-cluster-
ro.xxxxx.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "myrepliace",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
 "myrepliace.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "my-cluster-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-001.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "my-cluster-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
```

```
                            "Address": "my-
cluster-002.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "my-cluster-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-003.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "AutomaticFailover": "disabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Modification du nombre de répliques dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DecreaseReplicaCountà la section Référence des AWS CLI commandes.

`delete-cache-cluster`

L'exemple de code suivant montre comment utiliser`delete-cache-cluster`.

AWS CLI

Pour supprimer un cluster de cache

L'`delete-cache-cluster`exemple suivant supprime le cluster précédemment provisionné spécifié. La commande supprime tous les nœuds de cache associés, les points de terminaison des nœuds et le cluster lui-même. Lorsque vous recevez une réponse satisfaisante suite à cette opération, Amazon commence ElastiCache immédiatement à supprimer le cluster ; vous ne pouvez ni annuler ni annuler cette opération.

Cette opération n'est pas valide dans les cas suivants :

Clusters Redis (mode cluster activé) Cluster qui est la dernière réplique lue d'un groupe de réplicationUn groupe de nœuds (partition) dont le mode multi-AZ est activéUn cluster issu d'un groupe de réplication Redis (mode cluster activé) Un cluster dont l'état n'est pas disponible

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id "my-cluster-002"
```

Sortie :

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster-002",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "deleting",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
        "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
        "PendingModifiedValues": {},
        "NotificationConfiguration": {
            "TopicArn": "arn:aws:sns:us-west-x:xxxxxxx4152:My_Topic",
            "TopicStatus": "active"
        },
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "mygroup",
```

```
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "kxkxk",
        "AutoMinorVersionUpgrade": true,
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-xxxxxxxxxx9836",
                "Status": "active"
            },
            {
                "SecurityGroupId": "sg-xxxxxxxxxxxx7b",
                "Status": "active"
            }
        ],
        "ReplicationGroupId": "my-cluster",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un cluster dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteCacheClusterà la section Référence des AWS CLI commandes.

`delete-cache-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-cache-parameter-group`.

AWS CLI

Pour supprimer un groupe de paramètres de cache

L'`delete-cache-parameter-group`exemple suivant supprime le groupe de paramètres de cache spécifié. Vous ne pouvez pas supprimer un groupe de paramètres de cache s'il est associé à un cluster de cache.

```
aws elasticache delete-cache-parameter-group \
```

```
    --cache-parameter-group-name myparamgroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un groupe de paramètres dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteCacheParameterGroupà la section Référence des AWS CLI commandes.

`delete-cache-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-cache-subnet-group`.

AWS CLI

Pour supprimer un groupe de sous-réseaux de cache

L'`delete-cache-subnet-group`exemple suivant supprime le groupe de sous-réseaux de cache spécifié. Vous ne pouvez pas supprimer un groupe de sous-réseaux de cache s'il est associé à des clusters.

```
aws elasticache delete-cache-subnet-group \
    --cache-subnet-group-name "mygroup"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un groupe de sous-réseaux dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteCacheSubnetGroupà la section Référence des AWS CLI commandes.

`delete-global-replication-group`

L'exemple de code suivant montre comment utiliser`delete-global-replication-group`.

AWS CLI

Pour supprimer un groupe de réplication global

L'`delete-global-replication-group`exemple suivant supprime un nouveau groupe de réplication global.

```
aws elasticache delete-global-replication-group \
    --global-replication-group-id my-global-replication-group \
    --retain-primary-replication-group
```

Sortie :

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-my-grg",
        "GlobalReplicationGroupDescription": "my-grg",
        "Status": "deleting",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-cluster-grg",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteGlobalReplicationGroupà la section Référence des AWS CLI commandes.

`delete-replication-group`

L'exemple de code suivant montre comment utiliser`delete-replication-group`.

AWS CLI

Pour supprimer un groupe de réplication

L'`delete-replication-group`exemple suivant supprime un groupe de réplication existant. Par défaut, cette opération supprime l'intégralité du groupe de réplication, y compris toutes primary/primaries les répliques lues. Si le groupe de réplication ne possède qu'un seul primaire, vous pouvez éventuellement supprimer uniquement les répliques en lecture, tout en conservant le principal en définissant RetainPrimaryCluster =true.

Lorsque vous recevez une réponse satisfaisante suite à cette opération, Amazon commence ElastiCache immédiatement à supprimer les ressources sélectionnées ; vous ne pouvez ni annuler ni annuler cette opération. Valable uniquement pour Redis.

```
aws elasticache delete-replication-group \
    --replication-group-id "mygroup"
```

Sortie :

```
{
     "ReplicationGroup": {
         "ReplicationGroupId": "mygroup",
         "Description": "my group",
         "Status": "deleting",
         "PendingModifiedValues": {},
         "AutomaticFailover": "disabled",
         "SnapshotRetentionLimit": 0,
         "SnapshotWindow": "06:00-07:00",
         "TransitEncryptionEnabled": false,
         "AtRestEncryptionEnabled": false
     }
 }
```

- Pour plus de détails sur l'API, reportez-vous DeleteReplicationGroupà la section Référence des AWS CLI commandes.

`delete-snapshot`

L'exemple de code suivant montre comment utiliser`delete-snapshot`.

AWS CLI

Suppression d'un instantané

L'`delete-snapshot`exemple suivant a supprimé un instantané à l'aide du moteur Redis.

```
aws elasticache delete-snapshot \
    --snapshot-name mysnapshot
```

Sortie :

```
{
    "Snapshot": {
        "SnapshotName": "my-cluster-snapshot",
        "ReplicationGroupId": "mycluster",
        "ReplicationGroupDescription": "mycluster",
        "SnapshotStatus": "deleting",
        "SnapshotSource": "manual",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "PreferredMaintenanceWindow": "thu:12:00-thu:13:00",
        "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxxxx152:My_Topic",
        "Port": 6379,
        "CacheParameterGroupName": "default.redis5.0.cluster.on",
        "CacheSubnetGroupName": "default",
        "VpcId": "vpc-a3e97cdb",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "13:00-14:00",
        "NumNodeGroups": 4,
        "AutomaticFailover": "enabled",
        "NodeSnapshots": [
            {
                "CacheClusterId": "mycluster-0002-003",
                "NodeGroupId": "0002",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2020-06-18T00:05:44.719000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            },
            {
                "CacheClusterId": "mycluster-0003-003",
```

```
                "NodeGroupId": "0003",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2019-12-05T19:13:15.912000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            },
            {
                "CacheClusterId": "mycluster-0004-002",
                "NodeGroupId": "0004",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2019-12-09T19:44:34.324000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            },
            {
                "CacheClusterId": "mycluster-0005-003",
                "NodeGroupId": "0005",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2020-06-18T00:05:44.775000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Backup and Restore ElastiCache for Redis dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteSnapshotà la section Référence des AWS CLI commandes.

`delete-user-group`

L'exemple de code suivant montre comment utiliser`delete-user-group`.

AWS CLI

Pour supprimer un groupe d'utilisateurs

L'`delete-user-group`exemple suivant supprime un groupe d'utilisateurs.

```
aws elasticache delete-user-group \
```

```
    --user-group-id myusergroup
```

Sortie :

```
{
    "UserGroupId": "myusergroup",
    "Status": "deleting",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteUserGroupà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

L'`delete-user`exemple suivant supprime un utilisateur.

```
aws elasticache delete-user \
    --user-id user2
```

Sortie :

```
{
    "UserId": "user1",
    "UserName": "myUser",
    "Status": "deleting",
    "Engine": "redis",
    "AccessString": "on ~* +@all",
```

```
    "UserGroupIds": [
        "myusergroup"
    ],
    "Authentication": {
        "Type": "password",
        "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`describe-cache-clusters`

L'exemple de code suivant montre comment utiliser`describe-cache-clusters`.

AWS CLI

Pour décrire un cluster de cache

L'`describe-cache-clusters`exemple suivant décrit un cluster de cache.

```
aws elasticache describe-cache-clusters
```

Sortie :

```
{
    "CacheClusters": [
    {
            "CacheClusterId": "my-cluster-003",
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/
elasticache/home#client-download:",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "CacheClusterStatus": "available",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
```

```
            "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
            "PendingModifiedValues": {},
            "NotificationConfiguration": {
                "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
                "TopicStatus": "active"
            },
            "CacheSecurityGroups": [],
            "CacheParameterGroup": {
                "CacheParameterGroupName": "default.redis5.0",
                "ParameterApplyStatus": "in-sync",
                "CacheNodeIdsToReboot": []
            },
            "CacheSubnetGroupName": "kxkxk",
            "AutoMinorVersionUpgrade": true,
            "SecurityGroups": [
                {
                    "SecurityGroupId": "sg-xxxxxd7b",
                    "Status": "active"
                }
            ],
            "ReplicationGroupId": "my-cluster",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "06:30-07:30",
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx152:cluster:my-cache-
cluster",
            "ReplicationGroupLogDeliveryEnabled": false,
            "LogDeliveryConfigurations": [
                {
                    "LogType": "slow-log",
                    "DestinationType": "cloudwatch-logs",
                    "DestinationDetails": {
                        "CloudWatchLogsDetails": {
                            "LogGroup": "test-log"
                        }
                    },
                    "LogFormat": "text",
                    "Status": "active"
                }
            ]
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Gestion des clusters dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeCacheClustersla section Référence des AWS CLI commandes.

`describe-cache-engine-versions`

L'exemple de code suivant montre comment utiliser`describe-cache-engine-versions`.

AWS CLI

Pour décrire une version du moteur de cache

L'`describe-cache-engine-versions`exemple suivant renvoie une liste des moteurs de cache disponibles et de leurs versions.

```
aws elasticache describe-cache-engine-versions \
    --engine "Redis"
```

Sortie :

```
{
    "CacheEngineVersions": [
        {
            "Engine": "redis",
            "EngineVersion": "2.6.13",
            "CacheParameterGroupFamily": "redis2.6",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.6.13"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.19",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.19"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.21",
```

```
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.21"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.22",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.22"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.23",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.23"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.24",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.24"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.6",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.6"
        },
        {
            "Engine": "redis",
            "EngineVersion": "3.2.10",
            "CacheParameterGroupFamily": "redis3.2",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 3.2.10"
        },
        {
            "Engine": "redis",
            "EngineVersion": "3.2.4",
            "CacheParameterGroupFamily": "redis3.2",
            "CacheEngineDescription": "Redis",
```

```
            "CacheEngineVersionDescription": "redis version 3.2.4"
        },
        {
            "Engine": "redis",
            "EngineVersion": "3.2.6",
            "CacheParameterGroupFamily": "redis3.2",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 3.2.6"
        },
        {
            "Engine": "redis",
            "EngineVersion": "4.0.10",
            "CacheParameterGroupFamily": "redis4.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 4.0.10"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.0",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.0"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.3",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.3"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.4",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.4"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.5"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeCacheEngineVersionsla section Référence des AWS CLI commandes.

`describe-cache-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-cache-parameter-groups`.

AWS CLI

Pour décrire un groupe de paramètres de cache

L'`describe-cache-parameter-groups`exemple suivant renvoie une liste de descriptions de groupes de paramètres de cache.

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name "mygroup"
```

Sortie :

```
{
    "CacheParameterGroups": [
        {
            "CacheParameterGroupName": "mygroup",
            "CacheParameterGroupFamily": "redis5.0",
            "Description": " "
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration des paramètres du moteur à l'aide de groupes de paramètres dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeCacheParameterGroupsla section Référence des AWS CLI commandes.

`describe-cache-parameters`

L'exemple de code suivant montre comment utiliser`describe-cache-parameters`.

AWS CLI

Pour décrire les paramètres du cache

L'exemple « describe-cache-parameters » suivant renvoie la liste détaillée des paramètres pour le groupe de paramètres de cache spécifié.

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name "myparamgroup"
```

Sortie :

```
{
    "Parameters": [
        {
            "ParameterName": "activedefrag",
            "ParameterValue": "yes",
            "Description": "Enabled active memory defragmentation",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-cycle-max",
            "ParameterValue": "75",
            "Description": "Maximal effort for defrag in CPU percentage",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-75",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-cycle-min",
            "ParameterValue": "5",
            "Description": "Minimal effort for defrag in CPU percentage",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-75",
```

```
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-ignore-bytes",
            "ParameterValue": "104857600",
            "Description": "Minimum amount of fragmentation waste to start active
defrag",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-max-scan-fields",
            "ParameterValue": "1000",
            "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-1000000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-threshold-lower",
            "ParameterValue": "10",
            "Description": "Minimum percentage of fragmentation to start active
defrag",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-threshold-upper",
            "ParameterValue": "100",
```

```
            "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "activerehashing",
            "ParameterValue": "yes",
            "Description": "Apply rehashing or not.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "appendfsync",
            "ParameterValue": "everysec",
            "Description": "fsync policy for AOF persistence",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "always,everysec,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "appendonly",
            "ParameterValue": "no",
            "Description": "Enable Redis persistence.",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-hard-limit",
```

```
            "ParameterValue": "0",
            "Description": "Normal client output buffer hard limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-limit",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
            "ParameterValue": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
```

```
            "ParameterValue": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
            "ParameterValue": "60",
            "Description": "Pubsub client output buffer soft limit in seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
            "ParameterValue": "60",
            "Description": "Replica client output buffer soft limit in seconds.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-query-buffer-limit",
            "ParameterValue": "1073741824",
            "Description": "Max size of a single client query buffer",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-1073741824",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "close-on-replica-write",
```

```
            "ParameterValue": "yes",
            "Description": "If enabled, clients who attempt to write to a read-only
replica will be disconnected. Applicable to 2.8.23 and higher.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "cluster-enabled",
            "ParameterValue": "no",
            "Description": "Enable cluster mode",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "cluster-require-full-coverage",
            "ParameterValue": "no",
            "Description": "Whether cluster becomes unavailable if one or more slots
are not covered",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "databases",
            "ParameterValue": "16",
            "Description": "Set the number of databases.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-1200000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
```

```
        {
            "ParameterName": "hash-max-ziplist-entries",
            "ParameterValue": "512",
            "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "hash-max-ziplist-value",
            "ParameterValue": "64",
            "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "hll-sparse-max-bytes",
            "ParameterValue": "3000",
            "Description": "HyperLogLog sparse representation bytes limit",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-16000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-eviction",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on evictions",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
```

```
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-expire",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on expired keys",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-server-del",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on key updates",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lfu-decay-time",
            "ParameterValue": "1",
            "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lfu-log-factor",
            "ParameterValue": "10",
            "Description": "The log factor for incrementing key counter for LFU
eviction policy",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-",
```

```
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "list-compress-depth",
            "ParameterValue": "0",
            "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "list-max-ziplist-size",
            "ParameterValue": "-2",
            "Description": "The number of entries allowed per internal list node can
be specified as a fixed maximum size or a maximum number of elements",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "-5,-4,-3,-2,-1,1-",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lua-replicate-commands",
            "ParameterValue": "yes",
            "Description": "Always enable Lua effect replication or not",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lua-time-limit",
            "ParameterValue": "5000",
```

```
            "Description": "Max execution time of a Lua script in milliseconds. 0
 for unlimited execution without warnings.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "5000",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "maxclients",
            "ParameterValue": "65000",
            "Description": "The maximum number of Redis clients.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "1-65000",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "maxmemory-policy",
            "ParameterValue": "volatile-lru",
            "Description": "Max memory policy.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "maxmemory-samples",
            "ParameterValue": "3",
            "Description": "Max memory samples.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
```

```
            "ParameterName": "min-replicas-max-lag",
            "ParameterValue": "10",
            "Description": "The maximum amount of replica lag in seconds beyond
which the master would stop taking writes. A value of 0 means the master always
takes writes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "min-replicas-to-write",
            "ParameterValue": "0",
            "Description": "The minimum number of replicas that must be present with
lag no greater than min-replicas-max-lag for master to take writes. Setting this to
0 means the master always takes writes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "notify-keyspace-events",
            "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
            "Source": "user",
            "DataType": "string",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "proto-max-bulk-len",
            "ParameterValue": "536870912",
            "Description": "Max size of a single element request",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-536870912",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
```

```
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "rename-commands",
            "ParameterValue": "",
            "Description": "Redis commands that can be dynamically renamed by the
customer",
            "Source": "user",
            "DataType": "string",
            "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.3",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "repl-backlog-size",
            "ParameterValue": "1048576",
            "Description": "The replication backlog size in bytes for PSYNC. This is
the size of the buffer which accumulates slave data when slave is disconnected for
some time, so that when slave reconnects again, only transfer the portion of data
which the slave missed. Minimum value is 16K.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "16384-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "repl-backlog-ttl",
            "ParameterValue": "3600",
            "Description": "The amount of time in seconds after the master no longer
have any slaves connected for the master to free the replication backlog. A value
of 0 means to never release the backlog.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-allow-chaining",
```

```
            "ParameterValue": "no",
            "Description": "Configures if chaining of replicas is allowed",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-ignore-maxmemory",
            "ParameterValue": "yes",
            "Description": "Determines if replica ignores maxmemory setting by not
evicting items independent from the master",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-lazy-flush",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous flushDB during replica sync",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "reserved-memory-percent",
            "ParameterValue": "25",
            "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
```

```
        },
        {
            "ParameterName": "set-max-intset-entries",
            "ParameterValue": "512",
            "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "slowlog-log-slower-than",
            "ParameterValue": "10000",
            "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "slowlog-max-len",
            "ParameterValue": "128",
            "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "stream-node-max-bytes",
            "ParameterValue": "4096",
            "Description": "The maximum size of a single node in a stream in bytes",
            "Source": "user",
```

```
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "stream-node-max-entries",
            "ParameterValue": "100",
            "Description": "The maximum number of items a single node in a stream
can contain",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "tcp-keepalive",
            "ParameterValue": "300",
            "Description": "If non-zero, send ACKs every given number of seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "timeout",
            "ParameterValue": "0",
            "Description": "Close connection if client is idle for a given number of
seconds, or never if 0.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0,20-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "zset-max-ziplist-entries",
            "ParameterValue": "128",
```

```
            "Description": "The maximum number of sorted set entries in order for
 the dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "zset-max-ziplist-value",
            "ParameterValue": "64",
            "Description": "The threshold of biggest sorted set entries in order for
 the dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des paramètres dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeCacheParameterslа section Référence des AWS CLI commandes.

`describe-cache-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-cache-subnet-groups`.

AWS CLI

Pour décrire les groupes de sous-réseaux de cache

L'`describe-cache-subnet-groups`exemple suivant renvoie une liste de groupes de sous-réseaux.

```
aws elasticache describe-cache-subnet-groups
```

Sortie :


```
{
    "CacheSubnetGroups": [
        {
            "CacheSubnetGroupName": "default",
            "CacheSubnetGroupDescription": "Default CacheSubnetGroup",
            "VpcId": "vpc-a3e97cdb",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-8d4bacf5",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-dde21380",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-6485ec4f",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-b4ebebff",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                }
            ]
        },
        {
            "CacheSubnetGroupName": "kxkxk",
            "CacheSubnetGroupDescription": "mygroup",
            "VpcId": "vpc-a3e97cdb",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-b4ebebff",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
```

```
                    }
                }
            ]
        },
        {
            "CacheSubnetGroupName": "test",
            "CacheSubnetGroupDescription": "test",
            "VpcId": "vpc-a3e97cdb",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-b4ebebff",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Sous-réseaux et groupes de sous-réseaux dans le guide de l'utilisateur d'Elasticache ou Sous-réseaux et groupes de sous-réseaux dans le guide de l'utilisateur pour Memcached. ElastiCache

- Pour plus de détails sur l'API, voir DescribeCacheSubnetGroupsla section Référence des AWS CLI commandes.

`describe-engine-default-parameters`

L'exemple de code suivant montre comment utiliser`describe-engine-default-parameters`.

AWS CLI

Pour décrire les paramètres par défaut du moteur

L'`describe-engine-default-parameters`exemple suivant renvoie le moteur par défaut et les informations de paramètres système pour le moteur de cache spécifié.

```
aws elasticache describe-engine-default-parameters \
    --cache-parameter-group-family "redis5.0"
```

Sortie :

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "activedefrag",
                "ParameterValue": "no",
                "Description": "Enabled active memory defragmentation",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-cycle-max",
                "ParameterValue": "75",
                "Description": "Maximal effort for defrag in CPU percentage",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-75",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-cycle-min",
                "ParameterValue": "5",
                "Description": "Minimal effort for defrag in CPU percentage",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-75",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-ignore-bytes",
                "ParameterValue": "104857600",
                "Description": "Minimum amount of fragmentation waste to start
 active defrag",
                "Source": "system",
                "DataType": "integer",
```

```
                "AllowedValues": "1048576-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-max-scan-fields",
                "ParameterValue": "1000",
                "Description": "Maximum number of set/hash/zset/list fields that
will be processed from the main dictionary scan",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-1000000",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-threshold-lower",
                "ParameterValue": "10",
                "Description": "Minimum percentage of fragmentation to start active
defrag",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-100",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-threshold-upper",
                "ParameterValue": "100",
                "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-100",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "activerehashing",
                "ParameterValue": "yes",
```

```
            "Description": "Apply rehashing or not.",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "appendfsync",
            "ParameterValue": "everysec",
            "Description": "fsync policy for AOF persistence",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "always,everysec,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "appendonly",
            "ParameterValue": "no",
            "Description": "Enable Redis persistence.",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-hard-limit",
            "ParameterValue": "0",
            "Description": "Normal client output buffer hard limit in bytes.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-limit",
            "ParameterValue": "0",
```

```
            "Description": "Normal client output buffer soft limit in bytes.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in seconds.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
            "ParameterValue": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
            "ParameterValue": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
            "ParameterValue": "60",
```

```
                "Description": "Pubsub client output buffer soft limit in seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
                "ParameterValue": "60",
                "Description": "Replica client output buffer soft limit in
 seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-query-buffer-limit",
                "ParameterValue": "1073741824",
                "Description": "Max size of a single client query buffer",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1048576-1073741824",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "close-on-replica-write",
                "ParameterValue": "yes",
                "Description": "If enabled, clients who attempt to write to a read-
only replica will be disconnected. Applicable to 2.8.23 and higher.",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
```

```
                "ParameterName": "cluster-enabled",
                "ParameterValue": "no",
                "Description": "Enable cluster mode",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "requires-reboot"
            },
            {
                "ParameterName": "cluster-require-full-coverage",
                "ParameterValue": "no",
                "Description": "Whether cluster becomes unavailable if one or more
slots are not covered",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "databases",
                "ParameterValue": "16",
                "Description": "Set the number of databases.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-1200000",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "requires-reboot"
            },
            {
                "ParameterName": "hash-max-ziplist-entries",
                "ParameterValue": "512",
                "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
```

```
            },
            {
                "ParameterName": "hash-max-ziplist-value",
                "ParameterValue": "64",
                "Description": "The threshold of biggest hash entries in order for
the dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "hll-sparse-max-bytes",
                "ParameterValue": "3000",
                "Description": "HyperLogLog sparse representation bytes limit",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-16000",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lazyfree-lazy-eviction",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous delete on evictions",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lazyfree-lazy-expire",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous delete on expired keys",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
```

```
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lazyfree-lazy-server-del",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous delete on key updates",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lfu-decay-time",
                "ParameterValue": "1",
                "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lfu-log-factor",
                "ParameterValue": "10",
                "Description": "The log factor for incrementing key counter for LFU
eviction policy",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "list-compress-depth",
                "ParameterValue": "0",
                "Description": "Number of quicklist ziplist nodes from each side
of the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
                "Source": "system",
```

```
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "list-max-ziplist-size",
                "ParameterValue": "-2",
                "Description": "The number of entries allowed per internal list node
can be specified as a fixed maximum size or a maximum number of elements",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "-5,-4,-3,-2,-1,1-",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lua-replicate-commands",
                "ParameterValue": "yes",
                "Description": "Always enable Lua effect replication or not",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lua-time-limit",
                "ParameterValue": "5000",
                "Description": "Max execution time of a Lua script in milliseconds.
0 for unlimited execution without warnings.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "5000",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "maxclients",
                "ParameterValue": "65000",
```

```
                "Description": "The maximum number of Redis clients.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-65000",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "requires-reboot"
            },
            {
                "ParameterName": "maxmemory-policy",
                "ParameterValue": "volatile-lru",
                "Description": "Max memory policy.",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "maxmemory-samples",
                "ParameterValue": "3",
                "Description": "Max memory samples.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "min-replicas-max-lag",
                "ParameterValue": "10",
                "Description": "The maximum amount of replica lag in seconds beyond
 which the master would stop taking writes. A value of 0 means the master always
 takes writes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
```

```
            {
                "ParameterName": "min-replicas-to-write",
                "ParameterValue": "0",
                "Description": "The minimum number of replicas that must be present
with lag no greater than min-replicas-max-lag for master to take writes. Setting
this to 0 means the master always takes writes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "notify-keyspace-events",
                "Description": "The keyspace events for Redis to notify Pub/Sub
clients about. By default all notifications are disabled",
                "Source": "system",
                "DataType": "string",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "proto-max-bulk-len",
                "ParameterValue": "536870912",
                "Description": "Max size of a single element request",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1048576-536870912",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "rename-commands",
                "ParameterValue": "",
                "Description": "Redis commands that can be dynamically renamed by
the customer",
                "Source": "system",
                "DataType": "string",
                "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
                "IsModifiable": true,
```

```
                "MinimumEngineVersion": "5.0.3",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "repl-backlog-size",
                "ParameterValue": "1048576",
                "Description": "The replication backlog size in bytes for PSYNC.
This is the size of the buffer which accumulates slave data when slave is
disconnected for some time, so that when slave reconnects again, only transfer the
portion of data which the slave missed. Minimum value is 16K.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "16384-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "repl-backlog-ttl",
                "ParameterValue": "3600",
                "Description": "The amount of time in seconds after the master no
longer have any slaves connected for the master to free the replication backlog. A
value of 0 means to never release the backlog.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "replica-allow-chaining",
                "ParameterValue": "no",
                "Description": "Configures if chaining of replicas is allowed",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "replica-ignore-maxmemory",
                "ParameterValue": "yes",
```

```
                "Description": "Determines if replica ignores maxmemory setting by
not evicting items independent from the master",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "replica-lazy-flush",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous flushDB during replica
sync",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "reserved-memory-percent",
                "ParameterValue": "25",
                "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-100",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "set-max-intset-entries",
                "ParameterValue": "512",
                "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
```

```
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "slowlog-log-slower-than",
                "ParameterValue": "10000",
                "Description": "The execution time, in microseconds, to exceed in
order for the command to get logged. Note that a negative number disables the slow
log, while a value of zero forces the logging of every command.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "slowlog-max-len",
                "ParameterValue": "128",
                "Description": "The length of the slow log. There is no limit to
this length. Just be aware that it will consume memory. You can reclaim memory used
by the slow log with SLOWLOG RESET.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "stream-node-max-bytes",
                "ParameterValue": "4096",
                "Description": "The maximum size of a single node in a stream in
bytes",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "stream-node-max-entries",
                "ParameterValue": "100",
```

```
                "Description": "The maximum number of items a single node in a
stream can contain",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "tcp-keepalive",
                "ParameterValue": "300",
                "Description": "If non-zero, send ACKs every given number of
seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "timeout",
                "ParameterValue": "0",
                "Description": "Close connection if client is idle for a given
number of seconds, or never if 0.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0,20-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "zset-max-ziplist-entries",
                "ParameterValue": "128",
                "Description": "The maximum number of sorted set entries in order
for the dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
```

```
            },
            {
                "ParameterName": "zset-max-ziplist-value",
                "ParameterValue": "64",
                "Description": "The threshold of biggest sorted set entries in order
 for the dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, voir DescribeEngineDefaultParameterslaSection Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour décrire les événements d'un groupe de réplication

L'`describe-events`exemple suivant renvoie une liste d'événements pour un groupe de réplication.

```
aws elasticache describe-events \
    --source-identifier test-cluster \
    --source-type replication-group
```

Sortie :

```
{
    "Events": [
        {
            "SourceIdentifier": "test-cluster",
            "SourceType": "replication-group",
```

```
            "Message": "Automatic failover has been turned on for replication group
 test-cluster",
            "Date": "2020-03-18T23:51:34.457Z"
        },
        {
            "SourceIdentifier": "test-cluster",
            "SourceType": "replication-group",
            "Message": "Replication group test-cluster created",
            "Date": "2020-03-18T23:50:31.378Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance des événements dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeEventsla section Référence des AWS CLI commandes.

`describe-global-replication-groups`

L'exemple de code suivant montre comment utiliser`describe-global-replication-groups`.

AWS CLI

Pour décrire les groupes de réplication globaux

L'`describe-global-replication-groups`exemple suivant renvoie les détails d'une banque de données globale.

```
aws elasticache describe-global-replication-groups \
    --global-replication-group-id my-grg
```

Sortie :

```
{
    "GlobalReplicationGroups": [
        {
            "GlobalReplicationGroupId": "my-grg",
            "GlobalReplicationGroupDescription": "my-grg",
            "Status": "creating",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
```

```
            "EngineVersion": "5.0.6",
            "ClusterEnabled": false,
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeGlobalReplicationGroupsla section Référence des AWS CLI commandes.

`describe-replication-groups`

L'exemple de code suivant montre comment utiliser`describe-replication-groups`.

AWS CLI

Pour renvoyer une liste des détails du groupe de réplication

L'`describe-replication-groups`exemple suivant renvoie les groupes de réplication.

```
aws elasticache describe-replication-groups
```

Sortie :

```
{
    "ReplicationGroups": [
        {
            "ReplicationGroupId": "my-cluster",
            "Description": "mycluster",
            "Status": "available",
            "PendingModifiedValues": {},
            "MemberClusters": [
                "pat-cluster-001",
                "pat-cluster-002",
                "pat-cluster-003",
                "pat-cluster-004"
            ],
            "NodeGroups": [
```

```
                {
                    "NodeGroupId": "0001",
                    "Status": "available",
                    "PrimaryEndpoint": {
                        "Address": "my-
cluster.xxxxih.ng.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "ReaderEndpoint": {
                        "Address": "my-cluster-
ro.xxxxih.ng.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "NodeGroupMembers": [
                        {
                            "CacheClusterId": "my-cluster-001",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-001.xxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "primary"
                        },
                        {
                            "CacheClusterId": "my-cluster-002",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-002.xxxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        },
                        {
                            "CacheClusterId": "my-cluster-003",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-003.xxxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
```

```
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        },
                        {
                            "CacheClusterId": "my-cluster-004",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-004.xxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        }
                    ]
                }
            ],
            "AutomaticFailover": "disabled",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "07:30-08:30",
            "ClusterEnabled": false,
            "CacheNodeType": "cache.r5.xlarge",
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxx152:replicationgroup:my-cluster",
            "LogDeliveryConfigurations": [
                {
                    "LogType": "slow-log",
                    "DestinationType": "cloudwatch-logs",
                    "DestinationDetails": {
                        "CloudWatchLogsDetails": {
                            "LogGroup": "test-log"
                        }
                    },
                    "LogFormat": "json",
                    "Status": "active"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeReplicationGroupsla section Référence des AWS CLI commandes.

`describe-reserved-cache-nodes-offerings`

L'exemple de code suivant montre comment utiliser`describe-reserved-cache-nodes-offerings`.

AWS CLI

Pour décrire reserved-cache-nodes-offerings

L'`describe-reserved-cache-nodes-offerings`exemple suivant renvoie les détails d'une reserved-cache-node option.

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Sortie :

```
{
    "ReservedCacheNodesOfferings": [
        {
            "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eacbcdc8e5",
            "CacheNodeType": "cache.t3.small",
            "Duration": 31536000,
            "FixedPrice": 97.0,
            "UsagePrice": 0.0,
            "ProductDescription": "memcached",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.011,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        },
        {
            "ReservedCacheNodesOfferingId": "0443a27b-4da5-4b90-b92d-929fbd7abed2",
            "CacheNodeType": "cache.m3.2xlarge",
```

```
            "Duration": 31536000,
            "FixedPrice": 1772.0,
            "UsagePrice": 0.0,
            "ProductDescription": "redis",
            "OfferingType": "Heavy Utilization",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.25,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        },

        ...

    ]
}
```

Pour plus d'informations, consultez Obtenir des informations sur les offres de nœuds réservés dans le guide de l'utilisateur d'Elasticache Redis ou Obtenir des informations sur les offres de nœuds réservés dans le guide de l'utilisateur d'Elasticache Memcached.

- Pour plus de détails sur l'API, voir DescribeReservedCacheNodesOfferingsla section Référence des AWS CLI commandes.

`describe-reserved-cache-nodes`

L'exemple de code suivant montre comment utiliser`describe-reserved-cache-nodes`.

AWS CLI

Pour décrire les nœuds de cache réservés

L'`describe-reserved-cache-nodes`exemple suivant renvoie des informations sur les nœuds de cache réservés pour ce compte ou sur le nœud de cache réservé spécifié.

était Elasticache describe-reserved-cache-nodes

Sortie :

```
{
    "ReservedCacheNodes": [
        {
```

```
            "ReservedCacheNodeId": "mynode",
            "ReservedCacheNodesOfferingId": "xxxxxxxxx-xxxxx-xxxxx-xxxx-xxxxxxxx71",
            "CacheNodeType": "cache.t3.small",
            "StartTime": "2019-12-06T02:50:44.003Z",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CacheNodeCount": 1,
            "ProductDescription": "redis",
            "OfferingType": "No Upfront",
            "State": "payment-pending",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.023,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ReservationARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxxx52:reserved-instance:mynode"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des coûts avec des nœuds réservés dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeReservedCacheNodesla section Référence des AWS CLI commandes.

`describe-service-updates`

L'exemple de code suivant montre comment utiliser`describe-service-updates`.

AWS CLI

Pour décrire les mises à jour de service

L'`describe-service-updates`exemple suivant renvoie des informations sur les mises à jour de service.

```
aws elasticache describe-service-updates
```

Sortie :

```
{
    "ServiceUpdates": [
        {
            "ServiceUpdateName": "elc-xxxxxxxx7-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateEndDate": "2020-02-09T15:59:59Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateDescription": "Upgrades to improve the security,
reliability, and operational performance of your ElastiCache nodes",
            "ServiceUpdateType": "security-update",
            "Engine": "redis, memcached",
            "EngineVersion": "redis 2.6.13 and onwards, memcached 1.4.5 and
onwards",
            "AutoUpdateAfterRecommendedApplyByDate": false,
            "EstimatedUpdateTime": "30 minutes per node"
        },
        {
            "ServiceUpdateName": "elc-xxxxxxxx4-001",
            "ServiceUpdateReleaseDate": "2019-06-11T15:00:00Z",
            "ServiceUpdateEndDate": "2019-10-01T09:24:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateRecommendedApplyByDate": "2019-07-11T14:59:59Z",
            "ServiceUpdateStatus": "expired",
            "ServiceUpdateDescription": "Upgrades to improve the security,
reliability, and operational performance of your ElastiCache nodes",
            "ServiceUpdateType": "security-update",
            "Engine": "redis",
            "EngineVersion": "redis 3.2.6, redis 4.0 and onwards",
            "AutoUpdateAfterRecommendedApplyByDate": false,
            "EstimatedUpdateTime": "30 minutes per node"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeServiceUpdatesla section Référence des AWS CLI commandes.

`describe-snapshots`

L'exemple de code suivant montre comment utiliser`describe-snapshots`.

AWS CLI

Pour décrire les instantanés

L'exemple « describe-snapshots » suivant renvoie des informations sur les instantanés de votre cluster ou de votre groupe de réplication.

```
aws elasticache describe-snapshots
```

Sortie :

```
{
    "Snapshots": [
        {
            "SnapshotName": "automatic.my-cluster2-002-2019-12-05-06-38",
            "CacheClusterId": "my-cluster2-002",
            "SnapshotStatus": "available",
            "SnapshotSource": "automated",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
            "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxx52:My_Topic",
            "Port": 6379,
            "CacheParameterGroupName": "default.redis5.0",
            "CacheSubnetGroupName": "kxkxk",
            "VpcId": "vpc-a3e97cdb",
            "AutoMinorVersionUpgrade": true,
            "SnapshotRetentionLimit": 1,
            "SnapshotWindow": "06:30-07:30",
            "NodeSnapshots": [
                {
                    "CacheNodeId": "0001",
                    "CacheSize": "5 MB",
                    "CacheNodeCreateTime": "2019-11-26T01:22:52.396Z",
                    "SnapshotCreateTime": "2019-12-05T06:38:23Z"
                }
            ]
        },
        {
```

```
            "SnapshotName": "myreplica-backup",
            "CacheClusterId": "myreplica",
            "SnapshotStatus": "available",
            "SnapshotSource": "manual",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-26T00:14:52.439Z",
            "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
            "Port": 6379,
            "CacheParameterGroupName": "default.redis5.0",
            "CacheSubnetGroupName": "kxkxk",
            "VpcId": "vpc-a3e97cdb",
            "AutoMinorVersionUpgrade": true,
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "09:00-10:00",
            "NodeSnapshots": [
                {
                    "CacheNodeId": "0001",
                    "CacheSize": "5 MB",
                    "CacheNodeCreateTime": "2019-11-26T00:14:52.439Z",
                    "SnapshotCreateTime": "2019-11-26T00:25:01Z"
                }
            ]
        },
        {
            "SnapshotName": "my-cluster",
            "CacheClusterId": "my-cluster-003",
            "SnapshotStatus": "available",
            "SnapshotSource": "manual",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-25T23:56:17.186Z",
            "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
            "Port": 6379,
            "CacheParameterGroupName": "default.redis5.0",
            "CacheSubnetGroupName": "kxkxk",
```

```
            "VpcId": "vpc-a3e97cdb",
            "AutoMinorVersionUpgrade": true,
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "09:00-10:00",
            "NodeSnapshots": [
                {
                    "CacheNodeId": "0001",
                    "CacheSize": "5 MB",
                    "CacheNodeCreateTime": "2019-11-25T23:56:17.186Z",
                    "SnapshotCreateTime": "2019-11-26T03:08:33Z"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Backup and Restore ElastiCache for Redis dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeSnapshotsla section Référence des AWS CLI commandes.

`describe-update-actions`

L'exemple de code suivant montre comment utiliser`describe-update-actions`.

AWS CLI

Pour décrire les actions de mise à jour

L'`describe-update-actions`exemple suivant renvoie les détails des actions de mise à jour.

```
aws elasticache describe-update-actions
```

Sortie :

```
{
    "UpdateActions": [
        {
            "ReplicationGroupId": "mycluster",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
```

```
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-12-05T19:15:19.995Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "9/9",
            "UpdateActionStatusModifiedDate": "2019-12-05T19:15:20.461Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        },
        {
            "CacheClusterId": "my-memcached-cluster",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-12-04T18:26:05.349Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "1/1",
            "UpdateActionStatusModifiedDate": "2019-12-04T18:26:05.352Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        },
        {
            "ReplicationGroupId": "my-cluster",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-11-26T03:36:26.320Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "4/4",
            "UpdateActionStatusModifiedDate": "2019-12-04T22:11:12.664Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        },
        {
            "ReplicationGroupId": "my-cluster2",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
```

```
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-11-26T01:26:01.617Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "3/3",
            "UpdateActionStatusModifiedDate": "2019-11-26T01:26:01.753Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        }
    ]
}
```

Pour plus d'informations, consultez les mises à jour en libre-service sur Amazon ElastiCache dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeUpdateActionsla section Référence des AWS CLI commandes.

`describe-user-groups`

L'exemple de code suivant montre comment utiliser`describe-user-groups`.

AWS CLI

Pour décrire les groupes d'utilisateurs

L'`describe-user-groups`exemple suivant renvoie une liste de groupes d'utilisateurs.

```
aws elasticache describe-user-groups
```

Sortie :

```
{
    "UserGroups": [
        {
            "UserGroupId": "myusergroup",
            "Status": "active",
            "Engine": "redis",
            "UserIds": [
                "default"
```

```
            ],
            "ReplicationGroups": [],
            "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
        }
    ]
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeUserGroupsla section Référence des AWS CLI commandes.

`describe-users`

L'exemple de code suivant montre comment utiliser`describe-users`.

AWS CLI

Pour décrire les utilisateurs

L'`describe-users`exemple suivant renvoie une liste d'utilisateurs.

```
aws elasticache describe-users
```

Sortie :

```
{
    "Users": [
        {
            "UserId": "default",
            "UserName": "default",
            "Status": "active",
            "Engine": "redis",
            "AccessString": "on ~* +@all",
            "UserGroupIds": [
                "myusergroup"
            ],
            "Authentication": {
                "Type": "no-password"
            },
```

```
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:default"
        },
        {
            "UserId": "user1",
            "UserName": "myUser",
            "Status": "active",
            "Engine": "redis",
            "AccessString": "on ~* +@all",
            "UserGroupIds": [],
            "Authentication": {
                "Type": "password",
                "PasswordCount": 1
            },
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
        },
        {
            "UserId": "user2",
            "UserName": "myUser",
            "Status": "active",
            "Engine": "redis",
            "AccessString": "on ~app::* -@all +@read +@hash +@bitmap +@geo -setbit -
bitfield -hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius
 -georadiusbymember",
            "UserGroupIds": [],
            "Authentication": {
                "Type": "password",
                "PasswordCount": 1
            },
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
        }
    ]
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DescribeUsersla section Référence des AWS CLI commandes.

`disassociate-global-replication-group`

L'exemple de code suivant montre comment utiliser`disassociate-global-replication-group`.

AWS CLI

Pour dissocier un cluster secondaire d'un groupe de réplication global

L'`disassociate-global-replication-group`exemple suivant supprime un cluster secondaire d'une banque de données globale

```
aws elasticache disassociate-global-replication-group \
    --global-replication-group-id my-grg \
    --replication-group-id my-cluster-grg-secondary \
    --replication-group-region us-east-1
```

Sortie :

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "my-grg",
        "GlobalReplicationGroupDescription": "my-grg",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-cluster-grg-secondary",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "my-cluster-grg",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
```

```
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir DisassociateGlobalReplicationGroupla section Référence des AWS CLI commandes.

`increase-node-groups-in-global-replication-group`

L'exemple de code suivant montre comment utiliser`increase-node-groups-in-global-replication-group`.

AWS CLI

Pour augmenter le nombre de groupes de nœuds dans un groupe de réplication global

Ce qui suit `increase-node-groups-in-global-replication-group` augmente le nombre de groupes de nœuds à l'aide du moteur Redis.

```
aws elasticache increase-node-groups-in-global-replication-group \
    --global-replication-group-id sgaui-pat-test-4 \
    --node-group-count 6 \
    --apply-immediately
```

Sortie :

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-test-4",
        "GlobalReplicationGroupDescription": "test-4",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-cluster-b",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
```

```
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "my-cluster-a",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-test-4-0001",
                "Slots": "0-234,2420-5461"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0002",
                "Slots": "5462-5904,6997-9830"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0003",
                "Slots": "10923-11190,13375-16383"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0004",
                "Slots": "235-2419,5905-6996"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0005",
                "Slots": "9831-10922,11191-13374"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir IncreaseNodeGroupsInGlobalReplicationGroupla section Référence des AWS CLI commandes.

`increase-replica-count`

L'exemple de code suivant montre comment utiliser`increase-replica-count`.

AWS CLI

Pour augmenter le nombre de répliques

L'`increase-replica-count`exemple suivant permet d'effectuer l'une des deux opérations suivantes. Il peut augmenter dynamiquement le nombre de répliques dans un groupe de réplication Redis (mode cluster désactivé). Il peut également augmenter dynamiquement le nombre de nœuds de réplication dans un ou plusieurs groupes de nœuds (partitions) d'un groupe de réplication Redis (mode cluster activé). Cette opération est effectuée sans interruption du cluster.

```
aws elasticache increase-replica-count \
    --replication-group-id "my-cluster" \
    --apply-immediately \
    --new-replica-count 3
```

Sortie :

```
{
     "ReplicationGroup": {
         "ReplicationGroupId": "my-cluster",
         "Description": " ",
         "Status": "modifying",
         "PendingModifiedValues": {},
         "MemberClusters": [
             "my-cluster-001",
             "my-cluster-002",
             "my-cluster-003",
             "my-cluster-004"
         ],
         "NodeGroups": [
             {
                 "NodeGroupId": "0001",
                 "Status": "modifying",
                 "PrimaryEndpoint": {
                     "Address": "my-
cluster.xxxxxih.ng.0001.usw2.cache.amazonaws.com",
                     "Port": 6379
                 },
```

```
                "ReaderEndpoint": {
                    "Address": "my-cluster-
ro.xxxxxxih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "my-cluster-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-001.xxxxxih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "my-cluster-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-003.xxxxxih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "AutomaticFailover": "disabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Augmenter le nombre de répliques dans un fragment dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir IncreaseReplicaCountla section Référence des AWS CLI commandes.

`list-allowed-node-type-modifications`

L'exemple de code suivant montre comment utiliser`list-allowed-node-type-modifications`.

AWS CLI

Pour répertorier les modifications de nœuds autorisées

L'`list-allowed-node-type-modifications`exemple suivant répertorie tous les types de nœuds disponibles auxquels vous pouvez adapter le type de nœud actuel de votre cluster Redis ou de votre groupe de réplication.

```
aws elasticache list-allowed-node-type-modifications \
    --replication-group-id "my-replication-group"
```

Sortie :

```
{
    "ScaleUpModifications": [
        "cache.m5.12xlarge",
        "cache.m5.24xlarge",
        "cache.m5.4xlarge",
        "cache.r5.12xlarge",
        "cache.r5.24xlarge",
        "cache.r5.2xlarge",
        "cache.r5.4xlarge"
    ],
    "ScaleDownModifications": [
        "cache.m3.large",
        "cache.m3.medium",
        "cache.m3.xlarge",
        "cache.m4.large",
        "cache.m4.xlarge",
        "cache.m5.2xlarge",
        "cache.m5.large",
        "cache.m5.xlarge",
        "cache.r3.large",
        "cache.r4.large",
        "cache.r4.xlarge",
```

```
        "cache.r5.large",
        "cache.t2.medium",
        "cache.t2.micro",
        "cache.t2.small",
        "cache.t3.medium",
        "cache.t3.micro",
        "cache.t3.small"
    ]
}
```

Pour plus d'informations, consultez la section Scaling ElastiCache for Redis Clusters dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ListAllowedNodeTypeModificationsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises d'une ressource.

```
aws elasticache list-tags-for-resource \
    --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "Project",
            "Value": "querySpeedUp"
        },
        {
            "Key": "Environment",
            "Value": "PROD"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Liste des balises à l'aide de la AWS CLI dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`modify-cache-cluster`

L'exemple de code suivant montre comment utiliser`modify-cache-cluster`.

AWS CLI

Pour modifier les clusters de cache

L'`modify-cache-cluster`exemple suivant modifie les paramètres du cluster spécifié.

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id "my-cluster" \
    --num-cache-nodes 1
```

Sortie :

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.m5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "available",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheClusterCreateTime": "2019-12-04T18:24:56.652Z",
        "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
        "PendingModifiedValues": {},
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "default.redis5.0",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
```

```
        },
        "CacheSubnetGroupName": "default",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:00-08:00",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Modification d'un ElastiCache cluster dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyCacheClusterla section Référence des AWS CLI commandes.

`modify-cache-parameter-group`

L'exemple de code suivant montre comment utilisermodify-cache-parameter-group.

AWS CLI

Pour modifier un groupe de paramètres de cache

L'modify-cache-parameter-groupexemple suivant modifie les paramètres du groupe de paramètres de cache spécifié.

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name "mygroup" \
    --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

Sortie :

```
{
    "CacheParameterGroupName": "mygroup"
}
```

Pour plus d'informations, consultez la section Modification d'un groupe de paramètres dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyCacheParameterGroupla section Référence des AWS CLI commandes.

`modify-cache-subnet-group`

L'exemple de code suivant montre comment utiliser`modify-cache-subnet-group`.

AWS CLI

Pour modifier un groupe de sous-réseaux de cache

L'`modify-cache-subnet-group`exemple suivant modifie le groupe de sous-réseaux de cache spécifié.

```
aws elasticache modify-cache-subnet-group \
    --cache-subnet-group-name kxkxk \
    --cache-subnet-group-description "mygroup"
```

Sortie :

```
{
    "CacheSubnetGroup": {
        "CacheSubnetGroupName": "kxkxk",
        "CacheSubnetGroupDescription": "mygroup",
        "VpcId": "vpc-xxxxcdb",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-xxxxbff",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Modification d'un groupe de sous-réseaux dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyCacheSubnetGroupla section Référence des AWS CLI commandes.

`modify-global-replication-group`

L'exemple de code suivant montre comment utiliser`modify-global-replication-group`.

AWS CLI

Pour modifier un groupe de réplication global

Ce qui suit `modify-global-replication-group` modifie les propriétés d'un groupe de réplication global, en désactivant dans ce cas le basculement automatique, à l'aide du moteur Redis.

```
aws elasticache modify-global-replication-group \
    --global-replication-group-id sgaui-pat-group \
    --apply-immediately \
    --no-automatic-failover-enabled
```

Sortie

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-test-group",
        "GlobalReplicationGroupDescription": " ",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Réplication entre AWS régions à l'aide d'une banque de données globale dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyGlobalReplicationGroupla section Référence des AWS CLI commandes.

`modify-replication-group-shard-configuration`

L'exemple de code suivant montre comment utiliser`modify-replication-group-shard-configuration`.

AWS CLI

Pour modifier la configuration d'une partition de groupe de réplication

Ce qui suit permet `modify-replication-group-shard-configuration` de réduire le nombre de groupes de nœuds à l'aide du moteur Redis.

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id mycluster \
    --node-group-count 3 \
    --apply-immediately \
    --node-groups-to-remove 0002
```

Sortie

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "mycluster",
        "Description": "mycluster",
        "GlobalReplicationGroupInfo": {},
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "mycluster-0002-001",
            "mycluster-0002-002",
            "mycluster-0002-003",
            "mycluster-0003-001",
            "mycluster-0003-002",
            "mycluster-0003-003",
            "mycluster-0003-004",
            "mycluster-0004-001",
            "mycluster-0004-002",
            "mycluster-0004-003",
            "mycluster-0005-001",
            "mycluster-0005-002",
            "mycluster-0005-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0002",
                "Status": "modifying",
                "Slots": "894-1767,3134-4443,5149-5461,6827-7332,12570-13662",
                "NodeGroupMembers": [
```

```
                    {
                        "CacheClusterId": "mycluster-0002-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    }
                ]
            },
            {
                "NodeGroupId": "0003",
                "Status": "modifying",
                "Slots":
"0-324,5462-5692,6784-6826,7698-8191,10923-11075,12441-12569,13663-16383",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0003-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-004",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
```

```
            },
            {
                "NodeGroupId": "0004",
                "Status": "modifying",
                "Slots": "325-336,4706-5148,7333-7697,9012-10922,11076-12440",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0004-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0004-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0004-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0005",
                "Status": "modifying",
                "Slots": "337-893,1768-3133,4444-4705,5693-6783,8192-9011",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0005-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0005-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0005-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    }
                ]
```

```
            }
        ],
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "ConfigurationEndpoint": {
            "Address": "mycluster.g2xbih.clustercfg.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "13:00-14:00",
        "ClusterEnabled": true,
        "CacheNodeType": "cache.r5.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Scaling ElastiCache for Redis Clusters dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyReplicationGroupShardConfigurationla section Référence des AWS CLI commandes.

`modify-replication-group`

L'exemple de code suivant montre comment utiliser`modify-replication-group`.

AWS CLI

Pour modifier un groupe de réplication

Ce qui suit `modify-replication-group` désactive le Multi-AZ à l'aide du moteur Redis.

```
aws elasticache modify-replication-group \
    --replication-group-id test-cluster \
    --no-multi-az-enabled \
    --apply-immediately
```

Sortie

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "test-cluster",
```

```
        "Description": "test-cluster",
        "GlobalReplicationGroupInfo": {
            "GlobalReplicationGroupId": "sgaui-pat-group",
            "GlobalReplicationGroupMemberRole": "PRIMARY"
        },
        "Status": "available",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "test-cluster-001",
            "test-cluster-002",
            "test-cluster-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "available",
                "PrimaryEndpoint": {
                    "Address": "test-
cluster.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "test-cluster-
ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "test-cluster-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "test-
cluster-001.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "test-cluster-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "test-
cluster-002.g2xbih.0001.usw2.cache.amazonaws.com",
```

```
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "test-cluster-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "test-
cluster-003.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "test-cluster-002",
        "AutomaticFailover": "enabled",
        "MultiAZ": "disabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "08:00-09:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Modification d'un groupe de réplication dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyReplicationGroupla section Référence des AWS CLI commandes.

`modify-user-group`

L'exemple de code suivant montre comment utiliser`modify-user-group`.

AWS CLI

Pour modifier un groupe d'utilisateurs

L'`modify-user-group`exemple suivant ajoute un utilisateur au groupe d'utilisateurs.

```
aws elasticache modify-user-group \
    --user-group-id myusergroup \
    --user-ids-to-add user1
```

Sortie :

```
{
    "UserGroupId": "myusergroup",
    "Status": "modifying",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "PendingChanges": {
        "UserIdsToAdd": [
            "user1"
        ]
    },
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyUserGroupla section Référence des AWS CLI commandes.

`modify-user`

L'exemple de code suivant montre comment utiliser`modify-user`.

AWS CLI

Pour modifier un utilisateur

L'`modify-user`exemple suivant modifie la chaîne d'accès d'un utilisateur.

```
aws elasticache modify-user \
    --user-id user2 \
    --append-access-string "on ~* +@all"
```

Sortie :

```
{
    "UserId": "user2",
    "UserName": "myUser",
    "Status": "modifying",
    "Engine": "redis",
    "AccessString": "on ~* +@all",
    "UserGroupIds": [],
    "Authentication": {
        "Type": "password",
        "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
}
```

Pour plus d'informations, consultez la section Authentification des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir ModifyUserla section Référence des AWS CLI commandes.

`purchase-reserved-cache-nodes-offering`

L'exemple de code suivant montre comment utiliser`purchase-reserved-cache-nodes-offering`.

AWS CLI

Pour acheter un reserved-cache-node-offering

L'`purchase-reserved-cache-nodes-offering`exemple suivant vous permet d'acheter une offre de nœuds de cache réservés.

```
aws elasticache purchase-reserved-cache-nodes-offering \
    --reserved-cache-nodes-offering-id xxxxxxx-4da5-4b90-b92d-929fbd7abed2
```

Sortie

```
{
    "ReservedCacheNode": {
        "ReservedCacheNodeId": "ri-2020-06-30-17-59-40-474",
        "ReservedCacheNodesOfferingId": "xxxxxxx-4da5-4b90-b92d-929fbd7abed2",
        "CacheNodeType": "cache.m3.2xlarge",
        "StartTime": "2020-06-30T17:59:40.474000+00:00",
        "Duration": 31536000,
        "FixedPrice": 1772.0,
        "UsagePrice": 0.0,
        "CacheNodeCount": 1,
        "ProductDescription": "redis",
        "OfferingType": "Heavy Utilization",
        "State": "payment-pending",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.25,
                "RecurringChargeFrequency": "Hourly"
            }
        ]
    }
}
```

Pour plus d'informations, consultez Obtenir des informations sur les offres de nœuds réservés dans le guide de l'utilisateur d'Elasticache Redis ou Obtenir des informations sur les offres de nœuds réservés dans le guide de l'utilisateur d'Elasticache Memcached.

- Pour plus de détails sur l'API, voir PurchaseReservedCacheNodesOfferingla section Référence des AWS CLI commandes.

`reboot-cache-cluster`

L'exemple de code suivant montre comment utiliser`reboot-cache-cluster`.

AWS CLI

Pour redémarrer un cluster de cache

L'`reboot-cache-cluster`exemple suivant redémarre une partie ou la totalité des nœuds de cache d'un cluster provisionné. Cette opération applique tous les groupes de paramètres de cache modifiés au cluster. L'opération de redémarrage a lieu dès que possible et entraîne une interruption momentanée du cluster. Au cours du redémarrage, l'état du cluster est défini sur`REBOOTING`.

```
aws elasticache reboot-cache-cluster \
    --cache-cluster-id "my-cluster-001" \
    --cache-node-ids-to-reboot "0001"
```

Sortie :

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster-001",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "rebooting cache cluster nodes",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
        "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
        "PendingModifiedValues": {},
        "NotificationConfiguration": {
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
            "TopicStatus": "active"
        },
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "mygroup",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "kxkxk",
        "AutoMinorVersionUpgrade": true,
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-xxxxxxxxxxxxx836",
                "Status": "active"
            },
            {
                "SecurityGroupId": "sg-xxxxxxxx7b",
                "Status": "active"
            }
        ],
        "ReplicationGroupId": "my-cluster",
```

```
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Redémarrage d'un cluster < https:// docs.aws.amazon.com/AmazonElastiCache/ latest/red-ug/Clusters .rebooting.html dans le guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir RebootCacheClusterla section Référence des AWS CLI commandes.

`reset-cache-parameter-group`

L'exemple de code suivant montre comment utiliser`reset-cache-parameter-group`.

AWS CLI

Pour réinitialiser un groupe de paramètres de cache

L'`reset-cache-parameter-group`exemple suivant modifie les paramètres d'un groupe de paramètres de cache selon la valeur par défaut du moteur ou du système. Vous pouvez réinitialiser des paramètres spécifiques en soumettant une liste de noms de paramètres. Pour réinitialiser l'ensemble du groupe de paramètres du cache, spécifiez les `--cache-parameter-group-name` paramètres `--reset-all-parameters` et.

```
aws elasticache reset-cache-parameter-group \
    --cache-parameter-group-name "mygroup" \
    --reset-all-parameters
```

Sortie :

```
{
    "CacheParameterGroupName": "mygroup"
}
```

- Pour plus de détails sur l'API, voir ResetCacheParameterGroupla section Référence des AWS CLI commandes.

`start-migration`

L'exemple de code suivant montre comment utiliser`start-migration`.

AWS CLI

Pour démarrer une migration

Ce qui suit permet de `start-migration` migrer vos données de Redis sur Amazon auto-hébergé vers EC2 Amazon ElastiCache, à l'aide du moteur Redis.

```
aws elasticache start-migration \
   --replication-group-id test \
   --customer-node-endpoint-
list "Address='test.g2xbih.ng.0001.usw2.cache.amazonaws.com',Port=6379"
```

Sortie

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "test",
        "Description": "test",
        "GlobalReplicationGroupInfo": {},
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "test-001",
            "test-002",
            "test-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "available",
                "PrimaryEndpoint": {
                    "Address": "test.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "test-ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
```

```
                    {
                        "CacheClusterId": "test-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
"test-001.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "test-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
"test-002.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "test-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
"test-003.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "test-002",
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
        "TransitEncryptionEnabled": false,
```

```
        "AtRestEncryptionEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Migration en ligne vers le ElastiCache guide de l'utilisateur d'Elasticache.

- Pour plus de détails sur l'API, voir StartMigrationla section Référence des AWS CLI commandes.

`test-failover`

L'exemple de code suivant montre comment utiliser`test-failover`.

AWS CLI

Pour tester le basculement d'un groupe de nœuds

L'`test-failover`exemple suivant teste le basculement automatique sur le groupe de nœuds spécifié (appelé partition dans la console) dans un groupe de réplication (appelé cluster dans la console).

```
aws elasticache test-failover /
    --replication-group-id "mycluster" /
    --node-group-id "0001"
```

Sortie :

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "mycluster",
        "Description": "My Cluster",
        "Status": "available",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "mycluster-0001-001",
            "mycluster-0001-002",
            "mycluster-0001-003",
            "mycluster-0002-001",
            "mycluster-0002-002",
            "mycluster-0002-003",
            "mycluster-0003-001",
```

```
            "mycluster-0003-002",
            "mycluster-0003-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "available",
                "Slots": "0-5461",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0001-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0001-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0001-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0002",
                "Status": "available",
                "Slots": "5462-10922",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0002-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-003",
                        "CacheNodeId": "0001",
```

```
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0003",
                "Status": "available",
                "Slots": "10923-16383",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0003-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    }
                ]
            }
        ],
        "AutomaticFailover": "enabled",
        "ConfigurationEndpoint": {
            "Address": "mycluster.xxxxih.clustercfg.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "13:00-14:00",
        "ClusterEnabled": true,
        "CacheNodeType": "cache.r5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

- Pour plus de détails sur l'API, voir TestFailoverla section Référence des AWS CLI commandes.

MediaStore exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with MediaStore.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-container`

L'exemple de code suivant montre comment utiliser`create-container`.

AWS CLI

Pour créer un conteneur

L'`create-container`exemple suivant crée un nouveau conteneur vide.

```
aws mediastore create-container --container-name ExampleContainer
```

Sortie :

```
{
    "Container": {
        "AccessLoggingEnabled": false,
        "CreationTime": 1563557265,
        "Name": "ExampleContainer",
        "Status": "CREATING",
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
    }
```

```
}
```

Pour plus d'informations, consultez la section Création d'un conteneur dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateContainerà la section Référence des AWS CLI commandes.

`delete-container-policy`

L'exemple de code suivant montre comment utiliser`delete-container-policy`.

AWS CLI

Pour supprimer une politique relative aux conteneurs

L'`delete-container-policy`exemple suivant supprime la politique attribuée au conteneur spécifié. Lorsque la politique est supprimée, AWS Elemental attribue MediaStore automatiquement la politique par défaut au conteneur.

```
aws mediastore delete-container-policy \
    --container-name LiveEvents
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteContainerPolicyla référence de l' MediaStore API AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteContainerPolicyà la section Référence des AWS CLI commandes.

`delete-container`

L'exemple de code suivant montre comment utiliser`delete-container`.

AWS CLI

Pour supprimer un conteneur

L'`delete-container`exemple suivant supprime le conteneur spécifié. Vous pouvez supprimer un conteneur uniquement s'il ne possède aucun objet.

```
aws mediastore delete-container \
    --container-name=ExampleLiveDemo
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un conteneur dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteContainerà la section Référence des AWS CLI commandes.

`delete-cors-policy`

L'exemple de code suivant montre comment utiliser`delete-cors-policy`.

AWS CLI

Pour supprimer une politique CORS

L'`delete-cors-policy`exemple suivant supprime la politique de partage de ressources entre origines (CORS) attribuée au conteneur spécifié.

```
aws mediastore delete-cors-policy \
    --container-name ExampleContainer
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer une politique CORS dans le Guide de l' MediaStore utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous DeleteCorsPolicyà la section Référence des AWS CLI commandes.

`delete-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`delete-lifecycle-policy`.

AWS CLI

Pour supprimer une politique de cycle de vie des objets

L'`delete-lifecycle-policy`exemple suivant supprime la politique de cycle de vie des objets attachée au conteneur spécifié. Cette modification peut prendre jusqu'à 20 minutes pour être prise en compte.

```
aws mediastore delete-lifecycle-policy \
    --container-name LiveEvents
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'une politique de cycle de vie d'objet dans le Guide de MediaStore l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous DeleteLifecyclePolicyà la section Référence des AWS CLI commandes.

`describe-container`

L'exemple de code suivant montre comment utiliser`describe-container`.

AWS CLI

Pour afficher les détails d'un conteneur

L'`describe-container`exemple suivant affiche les détails du conteneur spécifié.

```
aws mediastore describe-container \
    --container-name ExampleContainer
```

Sortie :

```
{
    "Container": {
        "CreationTime": 1563558086,
        "AccessLoggingEnabled": false,
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
        "Status": "ACTIVE",
        "Name": "ExampleContainer",
        "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com"
    }
```

```
}
```

Pour plus d'informations, consultez la section Affichage des détails d'un conteneur dans le Guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribeContainerà la section Référence des AWS CLI commandes.

`describe-object`

L'exemple de code suivant montre comment utiliser`describe-object`.

AWS CLI

Pour afficher la liste des objets et des dossiers d'un conteneur spécifique

L'`describe-object`exemple suivant montre comment afficher les éléments (objets et dossiers) stockés dans un conteneur spécifique.

```
aws mediastore-data describe-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/file1234.jpg
```

Sortie :

```
{
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Pour plus d'informations, consultez la section Affichage des détails d'un objet dans le Guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribeObjectà la section Référence des AWS CLI commandes.

`get-container-policy`

L'exemple de code suivant montre comment utiliser`get-container-policy`.

AWS CLI

Pour consulter une politique relative aux conteneurs

L'`get-container-policy`exemple suivant montre la politique basée sur les ressources du conteneur spécifié.

```
aws mediastore get-container-policy \
    --container-name ExampleLiveDemo
```

Sortie :

```
{
    "Policy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "PublicReadOverHttps",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:root"
                },
                "Action": [
                    "mediastore:GetObject",
                    "mediastore:DescribeObject"
                ],
                "Resource": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo/",
                "Condition": {
                    "Bool": {
                        "aws:SecureTransport": "true"
                    }
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Affichage d'une politique de conteneur dans le Guide de MediaStore l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous GetContainerPolicyà la section Référence des AWS CLI commandes.

`get-cors-policy`

L'exemple de code suivant montre comment utiliser`get-cors-policy`.

AWS CLI

Pour consulter une politique CORS

L'`get-cors-policy`exemple suivant montre la politique de partage de ressources entre origines (CORS) attribuée au conteneur spécifié.

```
aws mediastore get-cors-policy \
    --container-name ExampleContainer \
    --region us-west-2
```

Sortie :

```
{
    "CorsPolicy": [
        {
            "AllowedMethods": [
                "GET",
                "HEAD"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                ""
            ],
            "AllowedHeaders": [
                ""
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une politique CORS dans le Guide de l' MediaStore utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous GetCorsPolicyà la section Référence des AWS CLI commandes.

`get-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`get-lifecycle-policy`.

AWS CLI

Pour consulter la politique de cycle de vie d'un objet

L'`get-lifecycle-policy`exemple suivant montre la politique de cycle de vie des objets attachée au conteneur spécifié.

```
aws mediastore get-lifecycle-policy \
    --container-name LiveEvents
```

Sortie :

```
{
    "LifecyclePolicy": {
        "rules": [
            {
                "definition": {
                    "path": [
                        {
                            "prefix": "Football/"
                        },
                        {
                            "prefix": "Baseball/"
                        }
                    ],
                    "days_since_create": [
                        {
                            "numeric": [
                                ">",
                                28
                            ]
                        }
                    ]
                },
                "action": "EXPIRE"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Affichage d'une politique de cycle de vie d'objet dans le Guide de MediaStore l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous GetLifecyclePolicyà la section Référence des AWS CLI commandes.

`get-object`

L'exemple de code suivant montre comment utiliser`get-object`.

AWS CLI

Pour télécharger un objet

L'`get-object`exemple suivant télécharge un objet vers le point de terminaison spécifié.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md README.md
```

Sortie :

```
{
    "ContentLength": "2307346",
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "StatusCode": 200
}
```

Pour télécharger une partie d'un objet

L'`get-object`exemple suivant télécharge une partie d'un objet vers le point de terminaison spécifié.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/README.md \
    --range="bytes=0-100" README2.md
```

Sortie :

```
{
    "StatusCode": 206,
    "ContentRange": "bytes 0-100/2307346",
    "ContentLength": "101",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentType": "image/jpeg",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Pour plus d'informations, consultez la section Téléchargement d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

`list-containers`

L'exemple de code suivant montre comment utiliser`list-containers`.

AWS CLI

Pour consulter la liste des conteneurs

L'`list-containers`exemple suivant affiche la liste de tous les conteneurs associés à votre compte.

```
aws mediastore list-containers
```

Sortie :

```
{
    "Containers": [
        {
            "CreationTime": 1505317931,
            "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
            "AccessLoggingEnabled": false,
            "Name": "ExampleLiveDemo"
```

```
        },
        {
            "CreationTime": 1506528818,
            "Endpoint": "https://fffggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
            "AccessLoggingEnabled": false,
            "Name": "ExampleContainer"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste de conteneurs dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListContainersà la section Référence des AWS CLI commandes.

`list-items`

L'exemple de code suivant montre comment utiliser`list-items`.

AWS CLI

Exemple 1 : pour afficher une liste d'objets et de dossiers dans un conteneur spécifique

L'`list-items`exemple suivant montre comment afficher les éléments (objets et dossiers) stockés dans le conteneur spécifié.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
            "Name": "filename.jpg",
```

```
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

Exemple 2 : pour afficher une liste d'objets et de dossiers dans un dossier spécifique

L'`list-items`exemple suivant montre comment afficher les éléments (objets et dossiers) stockés dans un dossier spécifique.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
            "Name": "filename.jpg",
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste d'objets dans le Guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListItemsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un conteneur

L'`list-tags-for-resource`exemple suivant affiche les clés de balise et les valeurs attribuées au conteneur spécifié.

```
aws mediastore list-tags-for-resource \
    --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

Sortie :

```
{
    "Tags": [
        {
            "Value": "Test",
            "Key": "Environment"
        },
        {
            "Value": "West",
            "Key": "Region"
        }
    ]
}
```

Pour plus d'informations, consultez ListTagsForResourcele manuel AWS Elemental MediaStore API Reference.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-container-policy`

L'exemple de code suivant montre comment utiliser`put-container-policy`.

AWS CLI

Pour modifier une politique de conteneur

L'`put-container-policy`exemple suivant attribue une politique différente au conteneur spécifié. Dans cet exemple, la politique mise à jour est définie dans un fichier nommé`LiveEventsContainerPolicy.json`.

```
aws mediastore put-container-policy \
    --container-name LiveEvents \
    --policy file://LiveEventsContainerPolicy.json
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modification d'une politique de conteneur dans le Guide de MediaStore l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous PutContainerPolicyà la section Référence des AWS CLI commandes.

`put-cors-policy`

L'exemple de code suivant montre comment utiliser`put-cors-policy`.

AWS CLI

Exemple 1 : pour ajouter une politique CORS

L'`put-cors-policy`exemple suivant ajoute une politique de partage de ressources entre origines (CORS) au conteneur spécifié. Le contenu de la politique CORS se trouve dans le fichier nommé`corsPolicy.json`.

```
aws mediastore put-cors-policy \
    --container-name ExampleContainer \
    --cors-policy file://corsPolicy.json
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Ajouter une politique CORS à un conteneur dans le Guide de l' MediaStore utilisateur AWS élémentaire.

Exemple 2 : pour modifier une politique CORS

L'`put-cors-policy`exemple suivant met à jour la politique de partage de ressources entre origines (CORS) attribuée au conteneur spécifié. Le contenu de la politique CORS mise à jour se trouve dans le fichier nommé`corsPolicy2.json`.

Pour plus d'informations, consultez la section Modification d'une politique CORS dans le Guide de l' MediaStore utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous PutCorsPolicyà la section Référence des AWS CLI commandes.

`put-lifecycle-policy`

L'exemple de code suivant montre comment utiliser`put-lifecycle-policy`.

AWS CLI

Pour créer une politique de cycle de vie des objets

L'`put-lifecycle-policy`exemple suivant attache une politique de cycle de vie des objets au conteneur spécifié. Cela vous permet de spécifier la durée pendant laquelle le service doit stocker les objets dans votre conteneur. MediaStore supprime les objets du conteneur une fois qu'ils ont atteint leur date d'expiration, comme indiqué dans la politique, qui figure dans le fichier nommé`LiveEventsLifecyclePolicy.json`.

```
aws mediastore put-lifecycle-policy \
    --container-name ExampleContainer \
    --lifecycle-policy file://ExampleLifecyclePolicy.json
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout d'une politique de cycle de vie d'objet à un conteneur dans le Guide de MediaStore l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous PutLifecyclePolicyà la section Référence des AWS CLI commandes.

`put-object`

L'exemple de code suivant montre comment utiliser`put-object`.

AWS CLI

Pour télécharger un objet

L'`put-object`exemple suivant télécharge un objet dans le conteneur spécifié. Vous pouvez spécifier le chemin du dossier dans lequel l'objet sera enregistré dans le conteneur. Si le dossier existe déjà, AWS Elemental y MediaStore stocke l'objet. Si le dossier n'existe pas, le service le crée, puis y stocke l'objet.

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body README.md \
    --path /folder_name/README.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

Sortie :

```
{
    "ContentSHA256":
 "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",
    "StorageClass": "TEMPORAL",
    "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"
}
```

Pour plus d'informations, consultez la section Chargement d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

`start-access-logging`

L'exemple de code suivant montre comment utiliser`start-access-logging`.

AWS CLI

Pour activer la journalisation des accès sur un conteneur

L'`start-access-logging`exemple suivant active la journalisation des accès sur le conteneur spécifié.

```
aws mediastore start-access-logging \
    --container-name LiveEvents
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de la journalisation des accès pour un conteneur dans le Guide de MediaStore l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous StartAccessLoggingà la section Référence des AWS CLI commandes.

`stop-access-logging`

L'exemple de code suivant montre comment utiliser`stop-access-logging`.

AWS CLI

Pour désactiver la journalisation des accès sur un conteneur

L'`stop-access-logging`exemple suivant désactive la journalisation des accès sur le conteneur spécifié.

```
aws mediastore stop-access-logging \
    --container-name LiveEvents
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation de la journalisation des accès pour un conteneur dans le Guide de l' MediaStore utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous StopAccessLoggingà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à un conteneur

L'`tag-resource`exemple suivant ajoute des clés et des valeurs de balise au conteneur spécifié.

```
aws mediastore tag-resource \
    --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer
 \
    --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value":
 "Test"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez TagResourcele manuel AWS Elemental MediaStore API Reference.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des étiquettes d'un conteneur

L'`untag-resource`exemple suivant supprime la clé de balise spécifiée et sa valeur associée d'un conteneur.

```
aws mediastore untag-resource \
    --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer
 \
    --tag-keys Region
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UntagResourcele manuel AWS Elemental MediaStore API Reference. .

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Exemples d'Amazon EMR utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon EMR.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-instance-fleet`

L'exemple de code suivant montre comment utiliser`add-instance-fleet`.

AWS CLI

Pour ajouter un parc d'instances de tâches à un cluster

Cet exemple ajoute un nouveau parc d'instances de tâches au cluster spécifié.

Commande :

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification='{Timeo
```

Sortie :

```
{
   "ClusterId": "j-12ABCDEFGHI34JK",
   "InstanceFleetId": "if-23ABCDEFGHI45JJ"
}
```

- Pour plus de détails sur l'API, reportez-vous AddInstanceFleetà la section Référence des AWS CLI commandes.

`add-steps`

L'exemple de code suivant montre comment utiliser`add-steps`.

AWS CLI

1. Pour ajouter des étapes JAR personnalisées à un cluster

Commande :

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-s3-demo-
bucket/mytest.jar,Args=arg1,arg2,arg3
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-s3-demo-
bucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3
```

Paramètres requis :

```
Jar
```

Paramètres facultatifs :

```
Type, Name, ActionOnFailure, Args
```

Sortie :

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

2. Pour ajouter des étapes de streaming à un cluster

Commande :

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=STREAMING,Name='Streaming
 Program',ActionOnFailure=CONTINUE,Args=[-files,s3://elasticmapreduce/samples/
wordcount/wordSplitter.py,-mapper,wordSplitter.py,-reducer,aggregate,-input,s3://
```

```
elasticmapreduce/samples/wordcount/input,-output,s3://amzn-s3-demo-bucket/wordcount/
output]
```

Paramètres requis :

```
Type, Args
```

Paramètres facultatifs :

```
Name, ActionOnFailure
```

Équivalent JSON (contenu de step.json) :

```
 [
  {
    "Name": "JSON Streaming Step",
    "Args": ["-files","s3://elasticmapreduce/samples/wordcount/wordSplitter.py","-
mapper","wordSplitter.py","-reducer","aggregate","-input","s3://elasticmapreduce/
samples/wordcount/input","-output","s3://amzn-s3-demo-bucket/wordcount/output"],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
  }
]
```

REMARQUE : les arguments JSON doivent inclure des options et des valeurs en tant qu'éléments propres dans la liste.

Commande (à l'aide de step.json) :

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file://./step.json
```

Sortie :

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

3. Pour ajouter une étape de streaming contenant plusieurs fichiers à un cluster (JSON uniquement)

JSON (multiplefiles.json) :

```
[
  {
     "Name": "JSON Streaming Step",
     "Type": "STREAMING",
     "ActionOnFailure": "CONTINUE",
     "Args": [
         "-files",
         "s3://amzn-s3-demo-bucket/mapper.py,s3://amzn-s3-demo-bucket/reducer.py",
         "-mapper",
         "mapper.py",
         "-reducer",
         "reducer.py",
         "-input",
         "s3://amzn-s3-demo-bucket/input",
         "-output",
         "s3://amzn-s3-demo-bucket/output"]
  }
]
```

Commande :

```
aws emr add-steps --cluster-id j-XXXXXXXX  --steps file://./multiplefiles.json
```

Paramètres requis :

```
Type, Args
```

Paramètres facultatifs :

```
Name, ActionOnFailure
```

Sortie :

```
{
    "StepIds":[
        "s-XXXXXXXX",
```

```
    ]
}
```

4. Pour ajouter des étapes Hive à un cluster

Commande :

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=HIVE,Name='Hive
 program',ActionOnFailure=CONTINUE,Args=[-f,s3://amzn-s3-demo-bucket/
myhivescript.q,-d,INPUT=s3://amzn-s3-demo-bucket/myhiveinput,-d,OUTPUT=s3://
amzn-s3-demo-bucket/myhiveoutput,arg1,arg2] Type=HIVE,Name='Hive
 steps',ActionOnFailure=TERMINATE_CLUSTER,Args=[-f,s3://elasticmapreduce/samples/
hive-ads/libs/model-build.q,-d,INPUT=s3://elasticmapreduce/samples/hive-ads/tables,-
d,OUTPUT=s3://amzn-s3-demo-bucket/hive-ads/output/2014-04-18/11-07-32,-d,LIBS=s3://
elasticmapreduce/samples/hive-ads/libs]
```

Paramètres requis :

```
Type, Args
```

Paramètres facultatifs :

```
Name, ActionOnFailure
```

Sortie :

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

5. Pour ajouter des étapes Pig à un cluster

Commande :

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=PIG,Name='Pig
 program',ActionOnFailure=CONTINUE,Args=[-f,s3://amzn-s3-demo-bucket/
mypigscript.pig,-p,INPUT=s3://amzn-s3-demo-bucket/mypiginput,-p,OUTPUT=s3://
```

```
amzn-s3-demo-bucket/mypigoutput,arg1,arg2] Type=PIG,Name='Pig program',Args=[-
f,s3://elasticmapreduce/samples/pig-apache/do-reports2.pig,-p,INPUT=s3://
elasticmapreduce/samples/pig-apache/input,-p,OUTPUT=s3://amzn-s3-demo-bucket/pig-
apache/output,arg1,arg2]
```

Paramètres requis :

```
Type, Args
```

Paramètres facultatifs :

```
Name, ActionOnFailure
```

Sortie :

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

6. Pour ajouter des étapes Impala à un cluster

Commande :

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=IMPALA,Name='Impala
 program',ActionOnFailure=CONTINUE,Args=--impala-script,s3://myimpala/input,--
console-output-path,s3://myimpala/output
```

Paramètres requis :

```
Type, Args
```

Paramètres facultatifs :

```
Name, ActionOnFailure
```

Sortie :

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AddStepsà la section Référence des AWS CLI commandes.

`add-tags`

L'exemple de code suivant montre comment utiliser`add-tags`.

AWS CLI

1. Pour ajouter des balises à un cluster

Commande :

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
 address="123 East NW Seattle"
```

Sortie :

```
None
```

2. Pour répertorier les balises d'un cluster

--Commande :

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

Sortie :

```
[
    {
        "Value": "male",
        "Key": "sex"
```

```
    },
    {
        "Value": "123 East NW Seattle",
        "Key": "address"
    },
    {
        "Value": "John Doe",
        "Key": "name"
    },
    {
        "Value": "29",
        "Key": "age"
    }
]
```

- Pour plus de détails sur l'API, reportez-vous AddTagsà la section Référence des AWS CLI commandes.

`create-cluster-examples`

L'exemple de code suivant montre comment utiliser`create-cluster-examples`.

AWS CLI

La plupart des exemples suivants supposent que vous avez spécifié votre rôle de service Amazon EMR et votre profil d' EC2 instance Amazon. Si vous ne l'avez pas encore fait, vous devez spécifier chaque rôle IAM requis ou utiliser le `--use-default-roles` paramètre lors de la création de votre cluster. Pour plus d'informations sur la spécification des rôles IAM, consultez Configurer les rôles IAM pour les autorisations Amazon EMR sur les services AWS dans le guide de gestion Amazon EMR.

Exemple 1 : pour créer un cluster

L'`create-cluster`exemple suivant crée un cluster EMR simple.

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --instance-type m4.large \
    --instance-count 2
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour créer un cluster Amazon EMR avec des rôles par défaut ServiceRole InstanceProfile

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise la `--instance-groups` configuration.

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Exemple 3 : pour créer un cluster Amazon EMR qui utilise un parc d'instances

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise la `--instance-fleets` configuration, en spécifiant deux types d'instances pour chaque flotte et deux EC2 sous-réseaux.

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
    --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m4.la
 InstanceFleetType=CORE,TargetSpotCapacity=11,InstanceTypeConfigs=['{InstanceType=m4.large,B
```

Exemple 4 : pour créer un cluster avec des rôles par défaut

L'`create-cluster`exemple suivant utilise le `--use-default-roles` paramètre pour spécifier le rôle de service et le profil d'instance par défaut.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --use-default-roles \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 5 : pour créer un cluster et spécifier les applications à installer

L'`create-cluster`exemple suivant utilise le `--applications` paramètre pour spécifier les applications installées par Amazon EMR. Cet exemple installe Hadoop, Hive et Pig.

```
aws emr create-cluster \
    --applications Name=Hadoop Name=Hive Name=Pig \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 6 : pour créer un cluster incluant Spark

L'exemple suivant installe Spark.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --applications Name=Spark \
    --ec2-attributes KeyName=myKey \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 7 : Pour spécifier une AMI personnalisée à utiliser pour les instances de cluster

L'`create-cluster`exemple suivant crée une instance de cluster basée sur l'AMI Amazon Linux avec IDami-a518e6df.

```
aws emr create-cluster \
    --name "Cluster with My Custom AMI" \
    --custom-ami-id ami-a518e6df \
    --ebs-root-volume-size 20 \
    --release-label emr-5.9.0 \
    --use-default-roles \
    --instance-count 2 \
    --instance-type m4.large
```

Exemple 8 : Pour personnaliser les configurations des applications

Les exemples suivants utilisent le `--configurations` paramètre pour spécifier un fichier de configuration JSON contenant des personnalisations d'applications pour Hadoop. Pour de plus

amples informations, veuillez consulter Configuration des applications dans le guide de version Amazon EMR.

Contenu de `configurations.json` :

```
[
    {
       "Classification": "mapred-site",
       "Properties": {
           "mapred.tasktracker.map.tasks.maximum": 2
       }
    },
    {
        "Classification": "hadoop-env",
        "Properties": {},
        "Configurations": [
            {
                "Classification": "export",
                "Properties": {
                    "HADOOP_DATANODE_HEAPSIZE": 2048,
                    "HADOOP_NAMENODE_OPTS": "-XX:GCTimeRatio=19"
                }
            }
        ]
    }
]
```

L'exemple suivant fait référence `configurations.json` à un fichier local.

```
aws emr create-cluster \
    --configurations file://configurations.json \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

L'exemple suivant fait référence `configurations.json` à un fichier dans Amazon S3.

```
aws emr create-cluster \
    --configurations https://s3.amazonaws.com/amzn-s3-demo-bucket/
configurations.json \
```

```
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 9 : pour créer un cluster avec des groupes d'instances principaux, principaux et de tâches

L'`create-cluster`exemple suivant permet `--instance-groups` de spécifier le type et le nombre d' EC2 instances à utiliser pour les groupes d'instances principaux, principaux et de tâches.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --instance-
groups Name=Master,InstanceGroupType=MASTER,InstanceType=m4.large,InstanceCount=1 Name=Core,
```

Exemple 10 : Pour spécifier qu'un cluster doit se terminer une fois toutes les étapes terminées

L'`create-cluster`exemple suivant indique que le cluster doit s'arrêter automatiquement une fois toutes les étapes terminées. `--auto-terminate`

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large
  InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.large \
    --auto-terminate
```

Exemple 11 : pour spécifier les détails de configuration du cluster tels que la paire de EC2 clés Amazon, la configuration réseau et les groupes de sécurité

L'`create-cluster`exemple suivant crée un cluster avec la paire de EC2 clés Amazon nommée `myKey` et un profil d'instance personnalisé nommé`myProfile`. Les paires de clés sont utilisées pour autoriser les connexions SSH aux nœuds du cluster, le plus souvent au nœud maître. Pour plus d'informations, consultez Utiliser une paire de EC2 clés Amazon pour les informations d'identification SSH dans le guide de gestion Amazon EMR.

```
aws emr create-cluster \
    --ec2-attributes KeyName=myKey,InstanceProfile=myProfile \
    --release-label emr-5.9.0 \
```

```
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

L'exemple suivant crée un cluster dans un sous-réseau Amazon VPC.

```
aws emr create-cluster \
    --ec2-attributes SubnetId=subnet-xxxxx \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

L'exemple suivant crée un cluster dans la zone de `us-east-1b` disponibilité.

```
aws emr create-cluster \
    --ec2-attributes AvailabilityZone=us-east-1b \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

L'exemple suivant crée un cluster et spécifie uniquement les groupes de sécurité gérés par Amazon EMR.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

L'exemple suivant crée un cluster et spécifie uniquement les groupes EC2 de sécurité Amazon supplémentaires.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
```

```
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

L'exemple suivant crée un cluster et spécifie les groupes de sécurité gérés par EMR, ainsi que des groupes de sécurité supplémentaires.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

L'exemple suivant crée un cluster dans un sous-réseau privé VPC et utilise un groupe de EC2 sécurité Amazon spécifique pour activer l'accès au service Amazon EMR, qui est requis pour les clusters situés dans des sous-réseaux privés.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,ServiceAccessSecurityGroup=sg-service-
access,EmrManagedMasterSecurityGroup=sg-master,EmrManagedSlaveSecurityGroup=sg-slave
 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

L'exemple suivant spécifie les paramètres de configuration du groupe de sécurité à l'aide d'un fichier JSON nommé `ec2_attributes.json` qui est stocké localement. REMARQUE : les arguments JSON doivent inclure des options et des valeurs en tant qu'éléments propres dans la liste.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes file://ec2_attributes.json  \
```

```
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Contenu de `ec2_attributes.json` :

```
[
    {
        "SubnetId": "subnet-xxxxx",
        "KeyName": "myKey",
        "InstanceProfile":"myRole",
        "EmrManagedMasterSecurityGroup": "sg-master1",
        "EmrManagedSlaveSecurityGroup": "sg-slave1",
        "ServiceAccessSecurityGroup": "sg-service-access",
        "AdditionalMasterSecurityGroups": ["sg-addMaster1","sg-addMaster2","sg-
addMaster3","sg-addMaster4"],
        "AdditionalSlaveSecurityGroups": ["sg-addSlave1","sg-addSlave2","sg-
addSlave3","sg-addSlave4"]
    }
]
```

Exemple 12 : Pour activer le débogage et spécifier un URI de journal

L'`create-cluster`exemple suivant utilise le `--enable-debugging` paramètre, qui vous permet d'afficher les fichiers journaux plus facilement à l'aide de l'outil de débogage de la console Amazon EMR. Le `--log-uri` paramètre est obligatoire avec`--enable-debugging`.

```
aws emr create-cluster \
    --enable-debugging \
    --log-uri s3://amzn-s3-demo-bucket/myLog \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 13 : Pour ajouter des balises lors de la création d'un cluster

Les balises sont des paires clé-valeur qui vous aident à identifier et à gérer les clusters. L'`create-cluster`exemple suivant utilise le `--tags` paramètre pour créer trois balises pour un cluster, une avec le nom de la clé `name` et la valeur`Shirley Rodriguez`, une seconde avec le nom de la clé `age` et la valeur29, et une troisième balise avec le nom de la clé `department` et la valeur`Analytics`.

```
aws emr create-cluster \
    --tags name="Shirley Rodriguez" age=29 department="Analytics" \
    --release-label emr-5.32.0 \
    --instance-type m5.xlarge \
    --instance-count 3 \
    --use-default-roles
```

L'exemple suivant répertorie les balises appliquées à un cluster.

```
aws emr describe-cluster \
    --cluster-id j-XXXXXXYY \
    --query Cluster.Tags
```

Exemple 14 : Pour utiliser une configuration de sécurité qui active le chiffrement et d'autres fonctionnalités de sécurité

L'`create-cluster`exemple suivant utilise le `--security-configuration` paramètre pour spécifier une configuration de sécurité pour un cluster EMR. Vous pouvez utiliser des configurations de sécurité avec Amazon EMR version 4.8.0 ou ultérieure.

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --security-configuration mySecurityConfiguration
```

Exemple 15 : pour créer un cluster avec des volumes de stockage EBS supplémentaires configurés pour les groupes d'instances

Lorsque vous spécifiez des volumes EBS supplémentaires, les arguments suivants sont requis :`VolumeType`, `SizeInGB` si `EbsBlockDeviceConfigs` est spécifié.

L'`create-cluster`exemple suivant crée un cluster avec plusieurs volumes EBS attachés aux EC2 instances du groupe d'instances principal.

```
aws emr create-cluster \
    --release-label emr-5.9.0  \
    --use-default-roles \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=d2.xlarge
 'InstanceGroupType=CORE,InstanceCount=2,InstanceType=d2.xlarge,EbsConfiguration={EbsOptimiz
```

```
{VolumeSpecification={VolumeType=io1,SizeInGB=100,Iops=100},VolumesPerInstance=4}]}'
 \
    --auto-terminate
```

L'exemple suivant crée un cluster avec plusieurs volumes EBS attachés aux EC2 instances du groupe d'instances principal.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --use-default-roles \
    --instance-groups 'InstanceGroupType=MASTER, InstanceCount=1,
 InstanceType=d2.xlarge, EbsConfiguration={EbsOptimized=true,
 EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=io1, SizeInGB=100,
 Iops=100}},
{VolumeSpecification={VolumeType=standard,SizeInGB=50},VolumesPerInstance=3}]}' InstanceGrou
 \
    --auto-terminate
```

Exemple 16 : Pour créer un cluster avec une politique de dimensionnement automatique

Vous pouvez associer des politiques de dimensionnement automatique aux groupes d'instances principaux et aux groupes d'instances de tâches à l'aide d'Amazon EMR version 4.0 et ultérieure. La politique de dimensionnement automatique ajoute et supprime des EC2 instances de manière dynamique en réponse à une CloudWatch métrique Amazon. Pour plus d'informations, consultez la section Utilisation du dimensionnement automatique dans Amazon EMR < https:// docs.aws.amazon.com/emr/ latest/ManagementGuide/emr -automatic-scaling.html>`_ dans le guide de gestion Amazon EMR.

Lorsque vous associez une politique de dimensionnement automatique, vous devez également spécifier le rôle par défaut pour le dimensionnement automatique à l'aide de`--auto-scaling-role EMR_AutoScaling_DefaultRole`.

L'`create-cluster`exemple suivant spécifie la politique de dimensionnement automatique pour le groupe d'`CORE`instances à l'aide de l'`AutoScalingPolicy`argument avec une structure JSON intégrée, qui spécifie la configuration de la politique de dimensionnement. Les groupes d'instances dotés d'une structure JSON intégrée doivent avoir l'ensemble des arguments entre guillemets simples. L'utilisation de guillemets simples est facultative pour les groupes d'instances sans structure JSON intégrée.

```
aws emr create-cluster
```

```
    --release-label emr-5.9.0 \
    --use-default-roles --auto-scaling-role EMR_AutoScaling_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceType=d2.xlarge,InstanceCount=1
 'InstanceGroupType=CORE,InstanceType=d2.xlarge,InstanceCount=2,AutoScalingPolicy={Constrain
```

L'exemple suivant utilise un fichier JSON pour spécifier la configuration de tous les groupes d'instances d'un cluster. `instancegroupconfig.json` Le fichier JSON spécifie la configuration de la politique de dimensionnement automatique pour le groupe d'instances principal.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-groups file://myfolder/instancegroupconfig.json \
    --auto-scaling-role EMR_AutoScaling_DefaultRole
```

Contenu de `instancegroupconfig.json` :

```
[
    {
        "InstanceCount": 1,
        "Name": "MyMasterIG",
        "InstanceGroupType": "MASTER",
        "InstanceType": "m4.large"
    },
    {
        "InstanceCount": 2,
        "Name": "MyCoreIG",
        "InstanceGroupType": "CORE",
        "InstanceType": "m4.large",
        "AutoScalingPolicy": {
            "Constraints": {
                "MinCapacity": 2,
                "MaxCapacity": 10
            },
            "Rules": [
                {
                    "Name": "Default-scale-out",
                    "Description": "Replicates the default scale-out rule in the
 console for YARN memory.",
                    "Action": {
```

```
                        "SimpleScalingPolicyConfiguration": {
                            "AdjustmentType": "CHANGE_IN_CAPACITY",
                            "ScalingAdjustment": 1,
                            "CoolDown": 300
                        }
                    },
                    "Trigger": {
                        "CloudWatchAlarmDefinition": {
                            "ComparisonOperator": "LESS_THAN",
                            "EvaluationPeriods": 1,
                            "MetricName": "YARNMemoryAvailablePercentage",
                            "Namespace": "AWS/ElasticMapReduce",
                            "Period": 300,
                            "Threshold": 15,
                            "Statistic": "AVERAGE",
                            "Unit": "PERCENT",
                            "Dimensions": [
                                {
                                    "Key": "JobFlowId",
                                    "Value": "${emr.clusterId}"
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
]
```

Exemple 17 : Ajouter des étapes JAR personnalisées lors de la création d'un cluster

L'`create-cluster`exemple suivant ajoute des étapes en spécifiant un fichier JAR stocké dans Amazon S3. Les étapes soumettent le travail à un cluster. La fonction principale définie dans le fichier JAR s'exécute après le provisionnement EC2 des instances, l'exécution de toutes les actions d'amorçage et l'installation des applications. Les étapes sont spécifiées à l'aide de`Type=CUSTOM_JAR`.

Les étapes JAR personnalisées nécessitent le `Jar=` paramètre, qui spécifie le chemin et le nom de fichier du JAR. Les paramètres facultatifs sont `TypeName`,`ActionOnFailure`,`Args`, et`MainClass`. Si aucune classe principale n'est spécifiée, le fichier JAR doit le spécifier `Main-Class` dans son fichier manifeste.

```
aws emr create-cluster \
    --steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-
s3-demo-bucket/
mytest.jar,Args=arg1,arg2,arg3 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s
amzn-s3-demo-bucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3  \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 18 : Pour ajouter des étapes de streaming lors de la création d'un cluster

Les `create-cluster` exemples suivants ajoutent une étape de streaming à un cluster qui s'arrête une fois toutes les étapes exécutées. Les étapes de streaming nécessitent des paramètres `Type` et`Args`. Les paramètres facultatifs des étapes de diffusion sont `Name` et`ActionOnFailure`.

L'exemple suivant indique l'étape en ligne.

```
aws emr create-cluster \
    --steps Type=STREAMING,Name='Streaming Program',ActionOnFailure=CONTINUE,Args=[-
files,s3://elasticmapreduce/samples/wordcount/wordSplitter.py,-
mapper,wordSplitter.py,-reducer,aggregate,-input,s3://elasticmapreduce/samples/
wordcount/input,-output,s3://amzn-s3-demo-bucket/wordcount/output] \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

L'exemple suivant utilise un fichier de configuration JSON stocké localement nommé`multiplefiles.json`. La configuration JSON spécifie plusieurs fichiers. Pour spécifier plusieurs fichiers au cours d'une étape, vous devez utiliser un fichier de configuration JSON pour spécifier l'étape. Les arguments JSON doivent inclure des options et des valeurs en tant qu'éléments propres dans la liste.

```
aws emr create-cluster \
    --steps file://./multiplefiles.json \
    --release-label emr-5.9.0  \
```

```
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Contenu de `multiplefiles.json` :

```
[
    {
        "Name": "JSON Streaming Step",
        "Args": [
            "-files",
            "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
            "-mapper",
            "wordSplitter.py",
            "-reducer",
            "aggregate",
            "-input",
            "s3://elasticmapreduce/samples/wordcount/input",
            "-output",
            "s3://amzn-s3-demo-bucket/wordcount/output"
        ],
        "ActionOnFailure": "CONTINUE",
        "Type": "STREAMING"
    }
]
```

Exemple 19 : Pour ajouter des étapes Hive lors de la création d'un cluster

L'exemple suivant ajoute les étapes Hive lors de la création d'un cluster. Les étapes de la ruche nécessitent des paramètres `Type` et. `Args` Les paramètres optionnels des étapes de la ruche sont `Name` et. `ActionOnFailure`

```
aws emr create-cluster \
    --steps Type=HIVE,Name='Hive
 program',ActionOnFailure=CONTINUE,ActionOnFailure=TERMINATE_CLUSTER,Args=[-
f,s3://elasticmapreduce/samples/hive-ads/libs/model-build.q,-d,INPUT=s3://
elasticmapreduce/samples/hive-ads/tables,-d,OUTPUT=s3://amzn-s3-demo-bucket/hive-
ads/output/2014-04-18/11-07-32,-d,LIBS=s3://elasticmapreduce/samples/hive-ads/libs]
 \
    --applications Name=Hive \
    --release-label emr-5.3.1 \
```

```
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Exemple 20 : Pour ajouter des étapes Pig lors de la création d'un cluster

L'exemple suivant ajoute des étapes Pig lors de la création d'un cluster. Les paramètres requis pour les étapes du porc sont `Type` et`Args`. Les paramètres optionnels de Pig Steps sont `Name` et`ActionOnFailure`.

```
aws emr create-cluster \
    --steps Type=PIG,Name='Pig program',ActionOnFailure=CONTINUE,Args=[-f,s3://
elasticmapreduce/samples/pig-apache/do-reports2.pig,-p,INPUT=s3://elasticmapreduce/
samples/pig-apache/input,-p,OUTPUT=s3://amzn-s3-demo-bucket/pig-apache/output] \
    --applications Name=Pig \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Exemple 21 : Pour ajouter des actions bootstrap

L'`create-cluster`exemple suivant exécute deux actions bootstrap définies comme des scripts stockés dans Amazon S3.

```
aws emr create-cluster \
    --bootstrap-actions Path=s3://amzn-s3-demo-bucket/
myscript1,Name=BootstrapAction1,Args=[arg1,arg2] Path=s3://amzn-s3-demo-bucket/
myscript2,Name=BootstrapAction2,Args=[arg1,arg2] \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

Exemple 22 : Pour activer la vue cohérente EMRFS et personnaliser les RetryCount paramètres et RetryPeriod

L'`create-cluster`exemple suivant indique le nombre de tentatives et la période de nouvelles tentatives pour une vue cohérente EMRFS. L'argument `Consistent=true` est obligatoire.

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
```

```
    --emrfs Consistent=true,RetryCount=6,RetryPeriod=30
```

L'exemple suivant spécifie la même configuration EMRFS que l'exemple précédent, en utilisant un fichier de configuration JSON stocké localement nommé. `emrfsconfig.json`

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --emrfs file://emrfsconfig.json
```

Contenu de `emrfsconfig.json` :

```
{
    "Consistent": true,
    "RetryCount": 6,
    "RetryPeriod": 30
}
```

Exemple 23 : Pour créer un cluster avec Kerberos configuré

Les `create-cluster` exemples suivants créent un cluster à l'aide d'une configuration de sécurité avec Kerberos activé, et établissent les paramètres Kerberos pour le cluster utilisant. `--kerberos-attributes`

La commande suivante spécifie les attributs Kerberos du cluster en ligne.

```
aws emr create-cluster \
    --instance-type m3.xlarge \
    --release-label emr-5.10.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --security-configuration mySecurityConfiguration \
    --kerberos-
attributes Realm=EC2.INTERNAL,KdcAdminPassword=123,CrossRealmTrustPrincipalPassword=123
```

La commande suivante spécifie les mêmes attributs, mais fait référence à un fichier JSON stocké localement nommé`kerberos_attributes.json`. Dans cet exemple, le fichier est enregistré dans le répertoire où vous exécutez la commande. Vous pouvez également référencer un fichier de configuration enregistré dans Amazon S3.

```
aws emr create-cluster \
```

```
    --instance-type m3.xlarge \
    --release-label emr-5.10.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --security-configuration mySecurityConfiguration \
    --kerberos-attributes file://kerberos_attributes.json
```

Contenu de `kerberos_attributes.json` :

```
{
    "Realm": "EC2.INTERNAL",
    "KdcAdminPassword": "123",
    "CrossRealmTrustPrincipalPassword": "123",
}
```

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise la `--instance-groups` configuration et dispose d'une politique de dimensionnement gérée.

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
    --managed-scaling-policy
 ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise le « --log-encryption-kms-key -id » pour définir l'ID de clé KMS utilisé pour le chiffrement des journaux.

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --log-uri s3://amzn-s3-demo-bucket/myLog \
    --log-encryption-kms-key-id arn:aws:kms:us-east-1:110302272565:key/
dd559181-283e-45d7-99d1-66da348c4d33 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise la configuration « --placement-group-configs » pour placer des nœuds maîtres dans un cluster à haute disponibilité (HA) au sein d'un groupe de placement à l'aide `SPREAD` d'une EC2 stratégie de placement.

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m4.largeInstanceGroupType=CORE,
 \
    --placement-group-configs InstanceRole=MASTER
```

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise la configuration « --auto-termination-policy » pour définir un seuil d'inactivité automatique pour le cluster.

```
aws emr create-cluster \
    --release-label emr-5.34.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-termination-policy IdleTimeout=100
```

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise le « -- os-release-label » pour définir une version Amazon Linux pour le lancement du cluster.

```
aws emr create-cluster \
    --release-label emr-6.6.0 \
    --os-release-label 2.0.20220406.1 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Exemple 24 : Pour spécifier les attributs d'un volume racine EBS : taille, E/S par seconde et débit pour les instances de cluster créées avec EMR versions 6.15.0 et ultérieures

L'`create-cluster`exemple suivant crée un cluster Amazon EMR qui utilise les attributs du volume racine pour configurer les spécifications des volumes racine pour les EC2 instances.

```
aws emr create-cluster \
    --name "Cluster with My Custom AMI" \
    --custom-ami-id ami-a518e6df \
```

```
    --ebs-root-volume-size 20 \
    --ebs-root-volume-iops 3000 \
    --ebs-root-volume-throughput 125 \
    --release-label emr-6.15.0 \
    --use-default-roles \
    --instance-count 2 \
    --instance-type m4.large
```

- Pour plus de détails sur l'API, reportez-vous CreateClusterExamplesà la section Référence des AWS CLI commandes.

`create-default-roles`

L'exemple de code suivant montre comment utiliser`create-default-roles`.

AWS CLI

1. Pour créer le rôle IAM par défaut pour EC2

   Commande :

   ```
   aws emr create-default-roles
   ```

   Sortie :

   ```
   If the role already exists then the command returns nothing.

   If the role does not exist then the output will be:

   [
       {
           "RolePolicy": {
               "Version": "2012-10-17",
               "Statement": [
                   {
                       "Action": [
                           "cloudwatch:*",
                           "dynamodb:*",
                           "ec2:Describe*",
                           "elasticmapreduce:Describe*",
                           "elasticmapreduce:ListBootstrapActions",
                           "elasticmapreduce:ListClusters",
   ```

```
                    "elasticmapreduce:ListInstanceGroups",
                    "elasticmapreduce:ListInstances",
                    "elasticmapreduce:ListSteps",
                    "kinesis:CreateStream",
                    "kinesis:DeleteStream",
                    "kinesis:DescribeStream",
                    "kinesis:GetRecords",
                    "kinesis:GetShardIterator",
                    "kinesis:MergeShards",
                    "kinesis:PutRecord",
                    "kinesis:SplitShard",
                    "rds:Describe*",
                    "s3:*",
                    "sdb:*",
                    "sns:*",
                    "sqs:*"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2008-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAIQ5SIQUGL5KMYBJX6",
        "CreateDate": "2015-06-09T17:09:04.602Z",
        "RoleName": "EMR_EC2_DefaultRole",
        "Path": "/",
        "Arn": "arn:aws:iam::176430881729:role/EMR_EC2_DefaultRole"
    }
},
{
```

```
        "RolePolicy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "ec2:AuthorizeSecurityGroupIngress",
                        "ec2:CancelSpotInstanceRequests",
                        "ec2:CreateSecurityGroup",
                        "ec2:CreateTags",
                        "ec2:DeleteTags",
                        "ec2:DescribeAvailabilityZones",
                        "ec2:DescribeAccountAttributes",
                        "ec2:DescribeInstances",
                        "ec2:DescribeInstanceStatus",
                        "ec2:DescribeKeyPairs",
                        "ec2:DescribePrefixLists",
                        "ec2:DescribeRouteTables",
                        "ec2:DescribeSecurityGroups",
                        "ec2:DescribeSpotInstanceRequests",
                        "ec2:DescribeSpotPriceHistory",
                        "ec2:DescribeSubnets",
                        "ec2:DescribeVpcAttribute",
                        "ec2:DescribeVpcEndpoints",
                        "ec2:DescribeVpcEndpointServices",
                        "ec2:DescribeVpcs",
                        "ec2:ModifyImageAttribute",
                        "ec2:ModifyInstanceAttribute",
                        "ec2:RequestSpotInstances",
                        "ec2:RunInstances",
                        "ec2:TerminateInstances",
                        "iam:GetRole",
                        "iam:GetRolePolicy",
                        "iam:ListInstanceProfiles",
                        "iam:ListRolePolicies",
                        "iam:PassRole",
                        "s3:CreateBucket",
                        "s3:Get*",
                        "s3:List*",
                        "sdb:BatchPutAttributes",
                        "sdb:Select",
                        "sqs:CreateQueue",
                        "sqs:Delete*",
                        "sqs:GetQueue*",
                        "sqs:ReceiveMessage"
```

```
                    ],
                    "Resource": "*",
                    "Effect": "Allow"
                }
            ]
        },
        "Role": {
            "AssumeRolePolicyDocument": {
                "Version": "2008-10-17",
                "Statement": [
                    {
                        "Action": "sts:AssumeRole",
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "elasticmapreduce.amazonaws.com"
                        }
                    }
                ]
            },
            "RoleId": "AROAI3SRVPPVSRDLARBPY",
            "CreateDate": "2015-06-09T17:09:10.401Z",
            "RoleName": "EMR_DefaultRole",
            "Path": "/",
            "Arn": "arn:aws:iam::176430881729:role/EMR_DefaultRole"
        }
    }
]
```

- Pour plus de détails sur l'API, reportez-vous CreateDefaultRolesà la section Référence des AWS CLI commandes.

`create-security-configuration`

L'exemple de code suivant montre comment utiliser`create-security-configuration`.

AWS CLI

1. Pour créer une configuration de sécurité avec le chiffrement en transit activé avec PEM pour le fournisseur de certificats, et le chiffrement au repos activé avec SSE-S3 pour le chiffrement S3 et AWS-KMS pour le fournisseur de clé de disque local

   Commande :

```
 aws emr create-security-configuration --name MySecurityConfig --security-
configuration '{
        "EncryptionConfiguration": {
                "EnableInTransitEncryption" : true,
                "EnableAtRestEncryption" : true,
                "InTransitEncryptionConfiguration" : {
                        "TLSCertificateConfiguration" : {
                                "CertificateProviderType" : "PEM",
                                "S3Object" : "s3://mycertstore/artifacts/
MyCerts.zip"
                        }
                },
                "AtRestEncryptionConfiguration" : {
                        "S3EncryptionConfiguration" : {
                                "EncryptionMode" : "SSE-S3"
                        },
                        "LocalDiskEncryptionConfiguration" : {
                                "EncryptionKeyProviderType" : "AwsKms",
                                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
                        }
                }
        }
}'
```

Sortie :

```
{
"CreationDateTime": 1474070889.129,
"Name": "MySecurityConfig"
}
```

Équivalent JSON (contenu de security_configuration.json) :

```
{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": true,
        "EnableAtRestEncryption": true,
        "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {
                "CertificateProviderType": "PEM",
                "S3Object": "s3://mycertstore/artifacts/MyCerts.zip"
```

```
            }
        },
        "AtRestEncryptionConfiguration": {
            "S3EncryptionConfiguration": {
                "EncryptionMode": "SSE-S3"
            },
            "LocalDiskEncryptionConfiguration": {
                "EncryptionKeyProviderType": "AwsKms",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
    }
}
```

Commande (à l'aide de security_configuration.json) :

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

Sortie :

```
{
"CreationDateTime": 1474070889.129,
"Name": "MySecurityConfig"
}
```

2. Pour créer une configuration de sécurité avec Kerberos activé à l'aide d'un KDC dédié au cluster et d'une confiance entre domaines

Commande :

```
 aws emr create-security-configuration --name MySecurityConfig --security-
configuration '{
     "AuthenticationConfiguration": {
         "KerberosConfiguration": {
             "Provider": "ClusterDedicatedKdc",
             "ClusterDedicatedKdcConfiguration": {
                 "TicketLifetimeInHours": 24,
                 "CrossRealmTrustConfiguration": {
                   "Realm": "AD.DOMAIN.COM",
                   "Domain": "ad.domain.com",
```

```
                    "AdminServer": "ad.domain.com",
                    "KdcServer": "ad.domain.com"
                }
            }
        }
    }
}'
```

Sortie :

```
{
"CreationDateTime": 1490225558.982,
"Name": "MySecurityConfig"
}
```

Équivalent JSON (contenu de security_configuration.json) :

```
{
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24,
                "CrossRealmTrustConfiguration": {
                    "Realm": "AD.DOMAIN.COM",
                    "Domain": "ad.domain.com",
                    "AdminServer": "ad.domain.com",
                    "KdcServer": "ad.domain.com"
                }
            }
        }
    }
}
```

Commande (à l'aide de security_configuration.json) :

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

Sortie :

```
{
```

```
"CreationDateTime": 1490225558.982,
"Name": "MySecurityConfig"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSecurityConfigurationà la section Référence des AWS CLI commandes.

`delete-security-configuration`

L'exemple de code suivant montre comment utiliser`delete-security-configuration`.

AWS CLI

Pour supprimer une configuration de sécurité dans la région actuelle

Commande :

```
aws emr delete-security-configuration --name MySecurityConfig
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteSecurityConfigurationà la section Référence des AWS CLI commandes.

`describe-cluster`

L'exemple de code suivant montre comment utiliser`describe-cluster`.

AWS CLI

Commande :

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

Sortie :

```
For release-label based uniform instance groups cluster:

        {
```

```
            "Cluster": {
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1436475075.199,
                        "CreationDateTime": 1436474656.563,
                    },
                    "State": "WAITING",
                    "StateChangeReason": {
                        "Message": "Waiting for steps to run"
                    }
                },
                "Ec2InstanceAttributes": {
                    "ServiceAccessSecurityGroup": "sg-xxxxxxxx",
                    "EmrManagedMasterSecurityGroup": "sg-xxxxxxxx",
                    "IamInstanceProfile": "EMR_EC2_DefaultRole",
                    "Ec2KeyName": "myKey",
                    "Ec2AvailabilityZone": "us-east-1c",
                    "EmrManagedSlaveSecurityGroup": "sg-yyyyyyyyy"
                },
                "Name": "My Cluster",
                "ServiceRole": "EMR_DefaultRole",
                "Tags": [],
                "TerminationProtected": true,
                "UnhealthyNodeReplacement": true,
                "ReleaseLabel": "emr-4.0.0",
                "NormalizedInstanceHours": 96,
                "InstanceGroups": [
                    {
                        "RequestedInstanceCount": 2,
                        "Status": {
                            "Timeline": {
                                "ReadyDateTime": 1436475074.245,
                                "CreationDateTime": 1436474656.564,
                                "EndDateTime": 1436638158.387
                            },
                            "State": "RUNNING",
                            "StateChangeReason": {
                                "Message": "",
                            }
                        },
                        "Name": "CORE",
                        "InstanceGroupType": "CORE",
                        "Id": "ig-YYYYYYY",
                        "Configurations": [],
```

```
                        "InstanceType": "m3.large",
                        "Market": "ON_DEMAND",
                        "RunningInstanceCount": 2
                    },
                    {
                        "RequestedInstanceCount": 1,
                        "Status": {
                            "Timeline": {
                                "ReadyDateTime": 1436475074.245,
                                "CreationDateTime": 1436474656.564,
                                "EndDateTime": 1436638158.387
                            },
                            "State": "RUNNING",
                            "StateChangeReason": {
                                "Message": "",
                            }
                        },
                        "Name": "MASTER",
                        "InstanceGroupType": "MASTER",
                        "Id": "ig-XXXXXXXXX",
                        "Configurations": [],
                        "InstanceType": "m3.large",
                        "Market": "ON_DEMAND",
                        "RunningInstanceCount": 1
                    }
                ],
                "Applications": [
                    {
                        "Name": "Hadoop"
                    }
                ],
                "VisibleToAllUsers": true,
                "BootstrapActions": [],
                "MasterPublicDnsName": "ec2-54-147-144-78.compute-1.amazonaws.com",
                "AutoTerminate": false,
                "Id": "j-XXXXXXXX",
                "Configurations": [
                    {
                        "Properties": {
                            "fs.s3.consistent.retryPeriodSeconds": "20",
                            "fs.s3.enableServerSideEncryption": "true",
                            "fs.s3.consistent": "false",
                            "fs.s3.consistent.retryCount": "2"
                        },
```

```
                        "Classification": "emrfs-site"
                    }
                ]
            }
        }


For release-label based instance fleet cluster:
{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1487897289.705,
                "CreationDateTime": 1487896933.942
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "EmrManagedMasterSecurityGroup": "sg-xxxxx",
            "RequestedEc2AvailabilityZones": [],
            "RequestedEc2SubnetIds": [],
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2AvailabilityZone": "us-east-1a",
            "EmrManagedSlaveSecurityGroup": "sg-xxxxx"
        },
        "Name": "My Cluster",
        "ServiceRole": "EMR_DefaultRole",
        "Tags": [],
        "TerminationProtected": false,
        "UnhealthyNodeReplacement": false,
        "ReleaseLabel": "emr-5.2.0",
        "NormalizedInstanceHours": 472,
        "InstanceCollectionType": "INSTANCE_FLEET",
        "InstanceFleets": [
            {
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1487897212.74,
                        "CreationDateTime": 1487896933.948
                    },
                    "State": "RUNNING",
```

```
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "ProvisionedSpotCapacity": 1,
                "Name": "MASTER",
                "InstanceFleetType": "MASTER",
                "LaunchSpecifications": {
                    "SpotSpecification": {
                        "TimeoutDurationMinutes": 60,
                        "TimeoutAction": "TERMINATE_CLUSTER"
                    }
                },
                "TargetSpotCapacity": 1,
                "ProvisionedOnDemandCapacity": 0,
                "InstanceTypeSpecifications": [
                    {
                        "BidPrice": "0.5",
                        "InstanceType": "m3.xlarge",
                        "WeightedCapacity": 1
                    }
                ],
                "Id": "if-xxxxxxx",
                "TargetOnDemandCapacity": 0
            }
        ],
        "Applications": [
            {
                "Version": "2.7.3",
                "Name": "Hadoop"
            }
        ],
        "ScaleDownBehavior": "TERMINATE_AT_INSTANCE_HOUR",
        "VisibleToAllUsers": true,
        "BootstrapActions": [],
        "MasterPublicDnsName": "ec2-xxx-xx-xxx-xx.compute-1.amazonaws.com",
        "AutoTerminate": false,
        "Id": "j-xxxxx",
        "Configurations": []
    }
}

For ami based uniform instance group cluster:
```

```
{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1399400564.432,
                "CreationDateTime": 1399400268.62
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2AvailabilityZone": "us-east-1c"
        },
        "Name": "My Cluster",
        "Tags": [],
        "TerminationProtected": true,
        "UnhealthyNodeReplacement": true,
        "RunningAmiVersion": "2.5.4",
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1399400558.848,
                        "CreationDateTime": 1399400268.621
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "Master instance group",
                "InstanceGroupType": "MASTER",
                "InstanceType": "m1.small",
                "Id": "ig-ABCD",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 2,
                "Status": {
```

```
                    "Timeline": {
                        "ReadyDateTime": 1399400564.439,
                        "CreationDateTime": 1399400268.621
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "Core instance group",
                "InstanceGroupType": "CORE",
                "InstanceType": "m1.small",
                "Id": "ig-DEF",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 2
            }
        ],
        "Applications": [
            {
                "Version": "1.0.3",
                "Name": "hadoop"
            }
        ],
        "BootstrapActions": [],
        "VisibleToAllUsers": false,
        "RequestedAmiVersion": "2.4.2",
        "LogUri": "s3://myLogUri/",
        "AutoTerminate": false,
        "Id": "j-XXXXXXXX"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeClusterà la section Référence des AWS CLI commandes.

`describe-step`

L'exemple de code suivant montre comment utiliser`describe-step`.

AWS CLI

La commande suivante décrit une étape avec l'ID d'étape `s-3LZC0QUT43AM` dans un cluster avec l'ID de cluster `j-3SD91U2E1L2QX` :

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

Sortie :

```
{
    "Step": {
        "Status": {
            "Timeline": {
                "EndDateTime": 1433200470.481,
                "CreationDateTime": 1433199926.597,
                "StartDateTime": 1433200404.959
            },
            "State": "COMPLETED",
            "StateChangeReason": {}
        },
        "Config": {
            "Args": [
                "s3://us-west-2.elasticmapreduce/libs/hive/hive-script",
                "--base-path",
                "s3://us-west-2.elasticmapreduce/libs/hive/",
                "--install-hive",
                "--hive-versions",
                "0.13.1"
            ],
            "Jar": "s3://us-west-2.elasticmapreduce/libs/script-runner/script-
runner.jar",
            "Properties": {}
        },
        "Id": "s-3LZC0QUT43AM",
        "ActionOnFailure": "TERMINATE_CLUSTER",
        "Name": "Setup hive"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeStepà la section Référence des AWS CLI commandes.

`get`

L'exemple de code suivant montre comment utiliser`get`.

AWS CLI

Ce qui suit télécharge l'`hadoop-examples.jar`archive depuis l'instance principale dans un cluster avec l'ID du cluster `j-3SD91U2E1L2QX` :

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /
home/hadoop-examples.jar --dest ~
```

- Pour plus de détails sur l'API, voir Get in AWS CLI Command Reference.

`list-clusters`

L'exemple de code suivant montre comment utiliser`list-clusters`.

AWS CLI

La commande suivante répertorie tous les clusters EMR actifs dans la région actuelle :

```
aws emr list-clusters --active
```

Sortie :

```
{
    "Clusters": [
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200405.353,
                    "CreationDateTime": 1433199926.596
                },
                "State": "WAITING",
                "StateChangeReason": {
                    "Message": "Waiting after step completed"
                }
            },
            "NormalizedInstanceHours": 6,
            "Id": "j-3SD91U2E1L2QX",
            "Name": "my-cluster"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListClustersà la section Référence des AWS CLI commandes.

`list-instance-fleets`

L'exemple de code suivant montre comment utiliser`list-instance-fleets`.

AWS CLI

Pour obtenir les détails de configuration des flottes d'instances d'un cluster

Cet exemple répertorie les détails des flottes d'instances du cluster spécifié.

Commande :

```
list-instance-fleets --cluster-id 'j-12ABCDEFGHI34JK'
```

Sortie :

```
{
  "InstanceFleets": [
      {
          "Status": {
              "Timeline": {
                  "ReadyDateTime": 1488759094.637,
                  "CreationDateTime": 1488758719.817
              },
              "State": "RUNNING",
              "StateChangeReason": {
                  "Message": ""
              }
          },
          "ProvisionedSpotCapacity": 6,
          "Name": "CORE",
          "InstanceFleetType": "CORE",
          "LaunchSpecifications": {
              "SpotSpecification": {
                  "TimeoutDurationMinutes": 60,
                  "TimeoutAction": "TERMINATE_CLUSTER"
              }
          },
          "ProvisionedOnDemandCapacity": 2,
```

```
            "InstanceTypeSpecifications": [
                {
                    "BidPrice": "0.5",
                    "InstanceType": "m3.xlarge",
                    "WeightedCapacity": 2
                }
            ],
            "Id": "if-1ABC2DEFGHIJ3"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759058.598,
                    "CreationDateTime": 1488758719.811
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 0,
            "Name": "MASTER",
            "InstanceFleetType": "MASTER",
            "ProvisionedOnDemandCapacity": 1,
            "InstanceTypeSpecifications": [
                {
                    "BidPriceAsPercentageOfOnDemandPrice": 100.0,
                    "InstanceType": "m3.xlarge",
                    "WeightedCapacity": 1
                }
            ],
           "Id": "if-2ABC4DEFGHIJ4"
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous ListInstanceFleetsà la section Référence des AWS CLI commandes.

`list-instances`

L'exemple de code suivant montre comment utiliser`list-instances`.

AWS CLI

La commande suivante répertorie toutes les instances d'un cluster avec l'ID du cluster `j-3C6XNQ39VR9WL` :

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

Sortie :

```
For a uniform instance group based cluster
  {
    "Instances": [
         {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200400.03,
                    "CreationDateTime": 1433199960.152
                },
                "State": "RUNNING",
                "StateChangeReason": {}
            },
            "Ec2InstanceId": "i-f19ecfee",
            "PublicDnsName": "ec2-52-52-41-150.us-west-2.compute.amazonaws.com",
            "PrivateDnsName": "ip-172-21-11-216.us-west-2.compute.internal",
            "PublicIpAddress": "52.52.41.150",
            "Id": "ci-3NNHQUQ2TWB6Y",
            "PrivateIpAddress": "172.21.11.216"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200400.031,
                    "CreationDateTime": 1433199949.102
                },
                "State": "RUNNING",
                "StateChangeReason": {}
            },
            "Ec2InstanceId": "i-1feee4c2",
            "PublicDnsName": "ec2-52-63-246-32.us-west-2.compute.amazonaws.com",
            "PrivateDnsName": "ip-172-31-24-130.us-west-2.compute.internal",
            "PublicIpAddress": "52.63.246.32",
            "Id": "ci-GAOCMKNKDCV7",
            "PrivateIpAddress": "172.21.11.215"
```

```
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200400.031,
                    "CreationDateTime": 1433199949.102
                },
                "State": "RUNNING",
                "StateChangeReason": {}
            },
            "Ec2InstanceId": "i-15cfeee3",
            "PublicDnsName": "ec2-52-25-246-63.us-west-2.compute.amazonaws.com",
            "PrivateDnsName": "ip-172-31-24-129.us-west-2.compute.internal",
            "PublicIpAddress": "52.25.246.63",
            "Id": "ci-2W3TDFFB47UAD",
            "PrivateIpAddress": "172.21.11.214"
        }
    ]
  }


For a fleet based cluster:
   {
      "Instances": [
          {
              "Status": {
                  "Timeline": {
                      "ReadyDateTime": 1487810810.878,
                      "CreationDateTime": 1487810588.367,
                      "EndDateTime": 1488022990.924
                  },
                  "State": "TERMINATED",
                  "StateChangeReason": {
                      "Message": "Instance was terminated."
                  }
              },
              "Ec2InstanceId": "i-xxxxx",
              "InstanceFleetId": "if-xxxxx",
              "EbsVolumes": [],
              "PublicDnsName": "ec2-xx-xxx-xxx-xxx.compute-1.amazonaws.com",
              "InstanceType": "m3.xlarge",
              "PrivateDnsName": "ip-xx-xx-xxx-xx.ec2.internal",
              "Market": "SPOT",
              "PublicIpAddress": "xx.xx.xxx.xxx",
```

```
            "Id": "ci-xxxxx",
            "PrivateIpAddress": "10.47.191.80"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListInstancesà la section Référence des AWS CLI commandes.

`list-security-configurations`

L'exemple de code suivant montre comment utiliser`list-security-configurations`.

AWS CLI

Pour répertorier les configurations de sécurité dans la région actuelle

Commande :

```
aws emr list-security-configurations
```

Sortie :

```
{
    "SecurityConfigurations": [
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-1"
        },
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-2"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSecurityConfigurationsà la section Référence des AWS CLI commandes.

`list-steps`

L'exemple de code suivant montre comment utiliser`list-steps`.

AWS CLI

La commande suivante répertorie toutes les étapes d'un cluster avec l'ID du cluster `j-3SD91U2E1L2QX` :

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- Pour plus de détails sur l'API, reportez-vous ListStepsà la section Référence des AWS CLI commandes.

`modify-cluster-attributes`

L'exemple de code suivant montre comment utilisermodify-cluster-attributes.

AWS CLI

La commande suivante définit la visibilité d'un cluster EMR avec l'ID `j-301CDNY0J5XM4` pour tous les utilisateurs :

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-
users
```

- Pour plus de détails sur l'API, reportez-vous ModifyClusterAttributesà la section Référence des AWS CLI commandes.

`modify-instance-fleet`

L'exemple de code suivant montre comment utilisermodify-instance-fleet.

AWS CLI

Pour modifier les capacités cibles d'un parc d'instances

Cet exemple modifie les capacités cibles On-Demand et Spot à 1 pour le parc d'instances spécifié.

Commande :

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- Pour plus de détails sur l'API, reportez-vous ModifyInstanceFleetà la section Référence des AWS CLI commandes.

`put`

L'exemple de code suivant montre comment utiliser`put`.

AWS CLI

La commande suivante télécharge un fichier nommé `healthcheck.sh` sur l'instance principale d'un cluster avec l'ID du cluster : `j-3SD91U2E1L2QX`

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/
scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- Pour plus de détails sur l'API, voir Put in AWS CLI Command Reference.

`remove-tags`

L'exemple de code suivant montre comment utiliser`remove-tags`.

AWS CLI

La commande suivante supprime une balise contenant la clé `prod` d'un cluster portant l'ID du cluster `j-3SD91U2E1L2QX` :

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- Pour plus de détails sur l'API, reportez-vous RemoveTagsà la section Référence des AWS CLI commandes.

`schedule-hbase-backup`

L'exemple de code suivant montre comment utiliser`schedule-hbase-backup`.

AWS CLI

Remarque : Cette commande ne peut être utilisée qu'avec HBase les versions 2.x et 3.x de l'AMI

1. Pour planifier une HBase sauvegarde complète **>>>>>>** 06ab6d6e13564b5733d75abaf3b599f93cf39a23

Commande :

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type full --dir
s3://amzn-s3-demo-bucket/backup --interval 10 --unit hours --start-time
2014-04-21T05:26:10Z --consistent
```

Sortie :

```
None
```

2. Pour planifier une sauvegarde incrémentielle HBase

Commande :

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type incremental
 --dir s3://amzn-s3-demo-bucket/backup --interval 30 --unit minutes --start-time
2014-04-21T05:26:10Z --consistent
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous ScheduleHbaseBackupà la section Référence des AWS CLI commandes.

`socks`

L'exemple de code suivant montre comment utiliser`socks`.

AWS CLI

La commande suivante ouvre une connexion Socks avec l'instance principale d'un cluster portant l'ID du cluster `j-3SD91U2E1L2QX` :

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

L'option de fichier de paires de clés utilise un chemin local vers un fichier de clé privée.

- Pour plus de détails sur l'API, consultez Socks dans AWS CLI Command Reference.

`ssh`

L'exemple de code suivant montre comment utiliser`ssh`.

AWS CLI

La commande suivante ouvre une connexion SSH avec l'instance principale d'un cluster portant l'ID du cluster : `j-3SD91U2E1L2QX`

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

L'option de fichier de paires de clés utilise un chemin local vers un fichier de clé privée.

Sortie :

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i /home/local/user/.ssh/
mykey.pem hadoop@ec2-52-52-41-150.us-west-2.compute.amazonaws.com
Warning: Permanently added 'ec2-52-52-41-150.us-
west-2.compute.amazonaws.com,52.52.41.150' (ECDSA) to the list of known hosts.
Last login: Mon Jun  1 23:15:38 2015

      __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
26 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.

--------------------------------------------------------------------------------

Welcome to Amazon Elastic MapReduce running Hadoop and Amazon Linux.

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.
```

```
The Hadoop UI can be accessed via the following commands:

  ResourceManager    lynx http://ip-172-21-11-216:9026/
  NameNode           lynx http://ip-172-21-11-216:9101/

------------------------------------------------------------------------------

[hadoop@ip-172-31-16-216 ~]$
```

- Pour plus de détails sur l'API, voir Ssh dans AWS CLI Command Reference.

Exemples d'Amazon EMR sur EKS utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon EMR sur EKS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`update-role-trust-policy`

L'exemple de code suivant montre comment utiliser`update-role-trust-policy`.

AWS CLI

Pour mettre à jour la politique de confiance d'un rôle IAM à utiliser avec Amazon EMR sur EKS

Cet exemple de commande met à jour la politique de confiance d'un rôle nommé example_iam_role afin qu'il puisse être utilisé avec Amazon EMR sur EKS avec l'espace de noms example_namespace issu d'un cluster EKS nommé example_cluster.

Commande :

```
aws emr-containers update-role-trust-policy \
    --cluster example_cluster \
    --namespace example_namespace \
    --role-name example_iam_role
```

Sortie :

```
If the trust policy has already been updated, then the output will be:
Trust policy statement already exists for role example_iam_role. No
changes were made!

If the trust policy has not been updated yet, then the output will be:
Successfully updated trust policy of role example_iam_role.
```

- Pour plus de détails sur l'API, voir UpdateRoleTrustPolicyla section Référence des AWS CLI commandes.

EventBridge exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with EventBridge.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-rule`

L'exemple de code suivant montre comment utiliser`delete-rule`.

AWS CLI

Pour supprimer une règle d' CloudWatch événements

Cet exemple supprime la règle nommée EC2 InstanceStateChanges :

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Pour plus de détails sur l'API, reportez-vous DeleteRuleà la section Référence des AWS CLI commandes.

`describe-rule`

L'exemple de code suivant montre comment utiliser`describe-rule`.

AWS CLI

Pour afficher les informations relatives à une règle d' CloudWatch événements

Cet exemple affiche des informations sur la règle nommée DailyLambdaFunction :

```
aws events describe-rule --name "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous DescribeRuleà la section Référence des AWS CLI commandes.

`disable-rule`

L'exemple de code suivant montre comment utiliser`disable-rule`.

AWS CLI

Pour désactiver une règle d' CloudWatch événements

Cet exemple désactive la règle nommée DailyLambdaFunction. La règle n’est pas supprimée :

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous DisableRuleà la section Référence des AWS CLI commandes.

`enable-rule`

L'exemple de code suivant montre comment utiliser`enable-rule`.

AWS CLI

Pour activer une règle d' CloudWatch événements

Cet exemple active la règle nommée DailyLambdaFunction, qui avait été précédemment désactivée :

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous EnableRuleà la section Référence des AWS CLI commandes.

`list-rule-names-by-target`

L'exemple de code suivant montre comment utiliser`list-rule-names-by-target`.

AWS CLI

Pour afficher toutes les règles ayant une cible spécifiée

Cet exemple affiche toutes les règles dont la cible est la fonction Lambda nommée MyFunctionName « » :

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-
east-1:123456789012:function:MyFunctionName"
```

- Pour plus de détails sur l'API, reportez-vous ListRuleNamesByTargetà la section Référence des AWS CLI commandes.

`list-rules`

L'exemple de code suivant montre comment utiliser`list-rules`.

AWS CLI

Pour afficher une liste de toutes les règles relatives aux CloudWatch événements

Cet exemple affiche toutes les règles relatives aux CloudWatch événements de la région :

```
aws events list-rules
```

Pour afficher une liste de règles relatives aux CloudWatch événements commençant par une certaine chaîne.

Cet exemple affiche toutes les règles relatives aux CloudWatch événements de la région dont le nom commence par « Quotidien » :

```
aws events list-rules --name-prefix "Daily"
```

- Pour plus de détails sur l'API, reportez-vous ListRulesà la section Référence des AWS CLI commandes.

`list-targets-by-rule`

L'exemple de code suivant montre comment utiliser`list-targets-by-rule`.

AWS CLI

Pour afficher toutes les cibles d'une règle d' CloudWatch événements

Cet exemple affiche toutes les cibles de la règle nommée DailyLambdaFunction :

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous ListTargetsByRuleà la section Référence des AWS CLI commandes.

`put-events`

L'exemple de code suivant montre comment utiliser`put-events`.

AWS CLI

Pour envoyer un événement personnalisé à CloudWatch Events

Cet exemple envoie un événement personnalisé à CloudWatch Events. L'événement est contenu dans le fichier putevents.json :

```
aws events put-events --entries file://putevents.json
```

Voici le contenu du fichier putevents.json :

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  },
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
   }
]
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS CLI commandes.

`put-rule`

L'exemple de code suivant montre comment utiliser`put-rule`.

AWS CLI

Pour créer des règles relatives aux CloudWatch événements

Cet exemple crée une règle qui se déclenche chaque jour à 9 h 00 (UTC). Si vous utilisez put-targets pour ajouter une fonction Lambda comme cible de cette règle, vous pouvez exécuter la fonction Lambda tous les jours à l'heure spécifiée :

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 *
 * ? *)"
```

Cet exemple crée une règle qui se déclenche lorsqu'une EC2 instance de la région change d'état :

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"source\":
[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change Notification\"]}"  --
role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Cet exemple crée une règle qui se déclenche lorsqu'une EC2 instance de la région est arrêtée ou terminée :

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-
pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change
 Notification\"],\"detail\":{\"state\":[\"stopped\",\"terminated\"]}}" --role-
arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS CLI commandes.

`put-targets`

L'exemple de code suivant montre comment utiliser`put-targets`.

AWS CLI

Pour ajouter des cibles aux règles relatives aux CloudWatch événements

Cet exemple ajoute une fonction Lambda comme cible d'une règle :

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Cet exemple définit un flux Amazon Kinesis comme cible, afin que les événements concernés par cette règle soient relayés vers le flux :

```
aws events put-targets --rule EC2InstanceStateChanges --targets
 "Id"="1","Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/
MyStream","RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Cet exemple définit deux flux Amazon Kinesis comme cibles pour une règle :

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="Target1","Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream1","RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda"
```

```
  "Id"="Target2"," Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream2","RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS CLI commandes.

`remove-targets`

L'exemple de code suivant montre comment utiliser`remove-targets`.

AWS CLI

Pour supprimer une cible pour un événement

Cet exemple supprime le flux Amazon Kinesis nommé MyStream 1 de la cible de la règle. DailyLambdaFunction Lors DailyLambdaFunction de sa création, ce flux a été défini comme cible avec l'ID Target1 :

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Pour plus de détails sur l'API, reportez-vous RemoveTargetsà la section Référence des AWS CLI commandes.

`test-event-pattern`

L'exemple de code suivant montre comment utiliser`test-event-pattern`.

AWS CLI

Pour vérifier si un modèle d'événement correspond à un événement spécifié

Cet exemple teste si le modèle « source:com.mycompany.myapp » correspond à l'événement spécifié. Dans cet exemple, le résultat serait « vrai » :

```
aws events test-event-pattern --event-pattern "{\"source\":[\"com.mycompany.myapp
\"]}" --event "{\"id\":\"1\",\"source\":\"com.mycompany.myapp\",\"detail-type\":
\"myDetailType\",\"account\":\"123456789012\",\"region\":\"us-east-1\",\"time\":
\"2017-04-11T20:11:04Z\"}"
```

- Pour plus de détails sur l'API, reportez-vous TestEventPatternà la section Référence des AWS CLI commandes.

EventBridge Exemples de tuyaux utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with EventBridge Pipes.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-pipe`

L'exemple de code suivant montre comment utiliser`create-pipe`.

AWS CLI

Pour créer un tuyau

L'`create-pipe`exemple suivant crée un canal nommé `Demo_Pipe` avec SQS comme source et CloudWatch Log Group comme cible pour le canal.

```
aws pipes create-pipe \
    --name Demo_Pipe \
    --desired-state RUNNING \
    --role-arn arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f \
    --source arn:aws:sqs:us-east-1:123456789012:Demo_Queue \
    --target arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/Demo_LogGroup
```

Sortie :

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
```

```
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "CREATING",
    "CreationTime": "2024-10-08T12:33:59-05:00",
    "LastModifiedTime": "2024-10-08T12:33:59.684839-05:00"
}
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreatePipeà la section Référence des AWS CLI commandes.

`delete-pipe`

L'exemple de code suivant montre comment utiliser`delete-pipe`.

AWS CLI

Pour supprimer un canal existant

L'`delete-pipe`exemple suivant supprime un Pipe nommé `Demo_Pipe` dans le compte spécifié.

```
aws pipes delete-pipe \
    --name Demo_Pipe
```

Sortie :

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "STOPPED",
    "CurrentState": "DELETING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T11:57:22-05:00"
}
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeletePipeà la section Référence des AWS CLI commandes.

`describe-pipe`

L'exemple de code suivant montre comment utiliser`describe-pipe`.

AWS CLI

Pour récupérer des informations sur un tuyau

L'`describe-pipe`exemple suivant affiche des informations sur le Pipe `Demo_Pipe` dans le compte spécifié.

```
aws pipes describe-pipe \
    --name Demo_Pipe
```

Sortie :

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "RUNNING",
    "StateReason": "User initiated",
    "Source": "arn:aws:sqs:us-east-1:123456789012:Demo_Queue",
    "SourceParameters": {
        "SqsQueueParameters": {
            "BatchSize": 1
        }
    },
    "EnrichmentParameters": {},
    "Target": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/
Demo_LogGroup",
    "TargetParameters": {},
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f",
    "Tags": {},
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T10:23:47-05:00",
    "LogConfiguration": {
        "CloudwatchLogsLogDestination": {
            "LogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
vendedlogs/pipes/Demo_Pipe"
        },
        "Level": "ERROR"
```

```
    }
}
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribePipeà la section Référence des AWS CLI commandes.

`list-pipes`

L'exemple de code suivant montre comment utiliser`list-pipes`.

AWS CLI

Pour récupérer une liste de tuyaux

L'`list-pipes`exemple suivant montre tous les canaux du compte spécifié.

```
aws pipes list-pipes
```

Sortie :

```
{
    "Pipes": [
        {
            "Name": "Demo_Pipe",
            "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
            "DesiredState": "RUNNING",
            "CurrentState": "RUNNING",
            "StateReason": "User initiated",
            "CreationTime": "2024-10-08T09:29:10-05:00",
            "LastModifiedTime": "2024-10-08T10:23:47-05:00",
            "Source": "arn:aws:sqs:us-east-1:123456789012:Demo_Queue",
            "Target": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/
Demo_LogGroup"
        }
    ]
}
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListPipesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises associées à un canal existant

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées à un canal nommé `Demo_Pipe` dans le compte spécifié.

```
aws pipes list-tags-for-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe
```

Sortie :

```
{
    "tags": {
        "stack": "Production",
        "team": "DevOps"
    }
}
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-pipe`

L'exemple de code suivant montre comment utiliser`start-pipe`.

AWS CLI

Pour démarrer un tuyau existant

L'`start-pipe`exemple suivant démarre un Pipe nommé `Demo_Pipe` dans le compte spécifié.

```
aws pipes start-pipe \
    --name Demo_Pipe
```

Sortie :

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "STARTING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T10:17:24-05:00"
}
```

Pour plus d'informations, consultez la section Démarrage ou arrêt d'un EventBridge canal Amazon dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartPipeà la section Référence des AWS CLI commandes.

`stop-pipe`

L'exemple de code suivant montre comment utiliser`stop-pipe`.

AWS CLI

Pour arrêter un tuyau existant

L'`stop-pipe`exemple suivant arrête un Pipe nommé `Demo_Pipe` dans le compte spécifié.

```
aws pipes stop-pipe \
    --name Demo_Pipe
```

Sortie :

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "STOPPED",
    "CurrentState": "STOPPING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
```

```
    "LastModifiedTime": "2024-10-08T09:29:49-05:00"
}
```

Pour plus d'informations, consultez la section Démarrage ou arrêt d'un EventBridge canal Amazon dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopPipeà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un canal existant

L'`tag-resource`exemple suivant balise un tuyau nomméDemo_Pipe. Si la commande réussit, aucune sortie n'est renvoyée.

```
aws pipes tag-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe \
    --tags stack=Production
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'un canal existant

L'`untag-resource`exemple suivant supprime une balise avec la clé `stack` du tuyau nomméDemo_Pipe. Si la commande réussit, aucune sortie n'est renvoyée.

```
aws pipes untag-resource \
```

```
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe \
    --tags stack
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-pipe`

L'exemple de code suivant montre comment utiliser`update-pipe`.

AWS CLI

Pour mettre à jour un canal existant

L'`update-pipe`exemple suivant met à jour le canal nommé `Demo_Pipe` en ajoutant un paramètre de configuration CloudWatch Log. Assurez-vous de mettre à jour le rôle d'exécution du canal afin qu'il dispose des autorisations appropriées pour la destination du journal.

```
aws pipes update-pipe \
    --name Demo_Pipe \
    --desired-state RUNNING \
    --log-configuration CloudwatchLogsLogDestination={LogGroupArn=arn:aws:logs:us-
east-1:123456789012:log-group:/aws/vendedlogs/pipes/Demo_Pipe},Level=TRACE \
    --role-arn arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f
```

Sortie :

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "UPDATING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T11:35:48-05:00"
}
```

Pour plus d'informations, consultez les concepts d'Amazon EventBridge Pipes dans le guide de EventBridge l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdatePipeà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Firewall Manager AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Firewall Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-admin-account`

L'exemple de code suivant montre comment utiliser`associate-admin-account`.

AWS CLI

Pour configurer le compte administrateur de Firewall Manager

L'`associate-admin-account`exemple suivant définit le compte administrateur pour Firewall Manager.

```
aws fms associate-admin-account \
    --admin-account 123456789012
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configurer le compte administrateur de AWS Firewall Manager dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous AssociateAdminAccountà la section Référence des AWS CLI commandes.

`delete-notification-channel`

L'exemple de code suivant montre comment utiliser`delete-notification-channel`.

AWS CLI

Pour supprimer les informations de la rubrique SNS pour les journaux de Firewall Manager

L'`delete-notification-channel`exemple suivant supprime les informations de la rubrique SNS.

```
aws fms delete-notification-channel
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configurer les notifications Amazon SNS et les CloudWatch alarmes Amazon dans le guide du AWS développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteNotificationChannelà la section Référence des AWS CLI commandes.

`delete-policy`

L'exemple de code suivant montre comment utiliser`delete-policy`.

AWS CLI

Pour supprimer une politique de Firewall Manager

L'`delete-policy`exemple suivant supprime la politique avec l'ID spécifié, ainsi que toutes ses ressources.

```
aws fms delete-policy \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --delete-all-policy-resources
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Working with AWS Firewall Manager Policies du AWS WAF, du AWS Firewall Manager et du AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`disassociate-admin-account`

L'exemple de code suivant montre comment utiliser`disassociate-admin-account`.

AWS CLI

Pour supprimer le compte administrateur de Firewall Manager

L'`disassociate-admin-account`exemple suivant supprime l'association actuelle du compte administrateur de Firewall Manager.

```
aws fms disassociate-admin-account
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configurer le compte administrateur de AWS Firewall Manager dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DisassociateAdminAccountà la section Référence des AWS CLI commandes.

`get-admin-account`

L'exemple de code suivant montre comment utiliser`get-admin-account`.

AWS CLI

Pour récupérer le compte administrateur de Firewall Manager

L'`get-admin-account`exemple suivant récupère le compte administrateur.

```
aws fms get-admin-account
```

Sortie :

```
{
    "AdminAccount": "123456789012",
    "RoleStatus": "READY"
}
```

Pour plus d'informations, consultez les conditions préalables àAWS Firewall Manager dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetAdminAccountà la section Référence des AWS CLI commandes.

`get-compliance-detail`

L'exemple de code suivant montre comment utiliser`get-compliance-detail`.

AWS CLI

Pour récupérer les informations de conformité d'un compte

L'`get-compliance-detail`exemple suivant récupère les informations de conformité pour la politique et le compte de membre spécifiés.

```
aws fms get-compliance-detail \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --member-account 123456789012
```

Sortie :

```
{
    "PolicyComplianceDetail": {
    "EvaluationLimitExceeded": false,
    "IssueInfoMap": {},
    "MemberAccount": "123456789012",
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyOwner": "123456789012",
    "Violators": []
}
```

Pour plus d'informations, consultez la section Visualisation de la conformité des ressources avec une politique dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetComplianceDetailà la section Référence des AWS CLI commandes.

`get-notification-channel`

L'exemple de code suivant montre comment utiliser`get-notification-channel`.

AWS CLI

Pour récupérer les informations des rubriques SNS pour les journaux de Firewall Manager

L'`get-notification-channel`exemple suivant permet de récupérer les informations de la rubrique SNS.

```
aws fms get-notification-channel
```

Sortie :

```
{
    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:us-west-2-fms",
    "SnsRoleName": "arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS"
}
```

Pour plus d'informations, consultez Configurer les notifications Amazon SNS et les CloudWatch alarmes Amazon dans le guide du AWS développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetNotificationChannelà la section Référence des AWS CLI commandes.

`get-policy`

L'exemple de code suivant montre comment utiliser`get-policy`.

AWS CLI

Pour récupérer une politique de Firewall Manager

L'`get-policy`exemple suivant récupère la politique avec l'ID spécifié.

```
aws fms get-policy \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "Policy": {
        "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "PolicyName": "test",
        "PolicyUpdateToken": "1:p+2RpKR4wPFx7mcrL1UOQQ==",
        "SecurityServicePolicyData": {
            "Type": "SECURITY_GROUPS_COMMON",
            "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_COMMON\",
\"revertManualSecurityGroupChanges\":true,\"exclusiveResourceSecurityGroupManagement
\":false,\"securityGroups\":[{\"id\":\"sg-045c43ccc9724e63e\"}]}"
        },
        "ResourceType": "AWS::EC2::Instance",
        "ResourceTags": [],
        "ExcludeResourceTags": false,
        "RemediationEnabled": false
    },
    "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/d1ac59b8-938e-42b3-
b2e0-7c620422ddc2"
}
```

Pour plus d'informations, consultez la section Working with AWS Firewall Manager Policies du AWS WAF, du AWS Firewall Manager et du AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetPolicyà la section Référence des AWS CLI commandes.

`list-compliance-status`

L'exemple de code suivant montre comment utiliser`list-compliance-status`.

AWS CLI

Pour récupérer les informations de conformité aux politiques pour les comptes des membres

L'`list-compliance-status`exemple suivant récupère les informations de conformité du compte membre pour la politique spécifiée.

```
aws fms list-compliance-status \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "PolicyComplianceStatusList": [
        {
            "PolicyOwner": "123456789012",
            "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "PolicyName": "test",
            "MemberAccount": "123456789012",
            "EvaluationResults": [
                {
                    "ComplianceStatus": "COMPLIANT",
                    "ViolatorCount": 0,
                    "EvaluationLimitExceeded": false
                },
                {
                    "ComplianceStatus": "NON_COMPLIANT",
                    "ViolatorCount": 2,
                    "EvaluationLimitExceeded": false
                }
            ],
            "LastUpdated": 1576283774.0,
            "IssueInfoMap": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation de la conformité des ressources avec une politique dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListComplianceStatusà la section Référence des AWS CLI commandes.

`list-member-accounts`

L'exemple de code suivant montre comment utiliser`list-member-accounts`.

AWS CLI

Pour récupérer les comptes des membres de l'organisation

L'`list-member-accounts`exemple suivant répertorie tous les comptes membres de l'organisation de l'administrateur de Firewall Manager.

```
aws fms list-member-accounts
```

Sortie :

```
{
    "MemberAccounts": [
        "222222222222",
        "333333333333",
        "444444444444"
    ]
}
```

Pour plus d'informations, consultez AWS Firewall Manager dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListMemberAccountsà la section Référence des AWS CLI commandes.

`list-policies`

L'exemple de code suivant montre comment utiliser`list-policies`.

AWS CLI

Pour récupérer toutes les politiques de Firewall Manager

L'`list-policies`exemple suivant permet de récupérer la liste des politiques du compte. Dans cet exemple, la sortie est limitée à deux résultats par demande. Chaque appel renvoie un `NextToken` qui peut être utilisé comme valeur du `--starting-token` paramètre lors du prochain `list-policies` appel afin d'obtenir le prochain ensemble de résultats pour la liste.

```
aws fms list-policies \
    --max-items 2
```

Sortie :

```
{
    "PolicyList": [
        {
            "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "PolicyName": "test",
            "ResourceType": "AWS::EC2::Instance",
            "SecurityServiceType": "SECURITY_GROUPS_COMMON",
            "RemediationEnabled": false
        },
        {
            "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "PolicyId": "457c9b21-fc94-406c-ae63-21217395ba72",
            "PolicyName": "test",
            "ResourceType": "AWS::EC2::Instance",
            "SecurityServiceType": "SECURITY_GROUPS_COMMON",
            "RemediationEnabled": false
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

Pour plus d'informations, consultez la section Working with AWS Firewall Manager Policies du AWS WAF, du AWS Firewall Manager et du AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListPoliciesà la section Référence des AWS CLI commandes.

`put-notification-channel`

L'exemple de code suivant montre comment utiliser`put-notification-channel`.

AWS CLI

Pour définir les informations de la rubrique SNS pour les journaux de Firewall Manager

L'`put-notification-channel`exemple suivant définit les informations de la rubrique SNS.

```
aws fms put-notification-channel \
```

```
    --sns-topic-arn arn:aws:sns:us-west-2:123456789012:us-west-2-fms \
    --sns-role-name arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configurer les notifications Amazon SNS et les CloudWatch alarmes Amazon dans le guide du AWS développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous PutNotificationChannelà la section Référence des AWS CLI commandes.

`put-policy`

L'exemple de code suivant montre comment utiliser`put-policy`.

AWS CLI

Pour créer une politique Firewall Manager

L'`put-policy`exemple suivant crée une politique de groupe de sécurité Firewall Manager.

```
aws fms put-policy \
    --cli-input-json file://policy.json
```

Contenu de `policy.json` :

```
{
    "Policy": {
        "PolicyName": "test",
        "SecurityServicePolicyData": {
            "Type": "SECURITY_GROUPS_USAGE_AUDIT",
            "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_USAGE_AUDIT\",
\"deleteUnusedSecurityGroups\":false,\"coalesceRedundantSecurityGroups\":true}"
        },
        "ResourceType": "AWS::EC2::SecurityGroup",
        "ResourceTags": [],
        "ExcludeResourceTags": false,
        "RemediationEnabled": false
    },
    "TagList": [
        {
```

```
            "Key": "foo",
            "Value": "foo"
        }
    ]
}
```

Sortie :

```
{
    "Policy": {
        "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "PolicyName": "test",
        "PolicyUpdateToken": "1:X9QGexP7HASDlsFp+G31Iw==",
        "SecurityServicePolicyData": {
            "Type": "SECURITY_GROUPS_USAGE_AUDIT",
            "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_USAGE_AUDIT\",
\"deleteUnusedSecurityGroups\":false,\"coalesceRedundantSecurityGroups\":true,
\"optionalDelayForUnusedInMinutes\":null}"
        },
        "ResourceType": "AWS::EC2::SecurityGroup",
        "ResourceTags": [],
        "ExcludeResourceTags": false,
        "RemediationEnabled": false
    },
    "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Working with AWS Firewall Manager Policies du AWS WAF, du AWS Firewall Manager et du AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutPolicyà la section Référence des AWS CLI commandes.

AWS FIS exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS FIS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-experiment-template`

L'exemple de code suivant montre comment utiliser`create-experiment-template`.

AWS CLI

Pour créer un modèle d'expérience

L'`create-experiment-template`exemple suivant crée un modèle d'expérience dans votre compte AWS FIS.

```
aws fis create-experiment-template \
    --cli-input-json file://myfile.json
```

Contenu de `myfile.json` :

```
{
    "description": "experimentTemplate",
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
        }
    ],
    "targets": {
        "Instances-Target-1": {
            "resourceType": "aws:ec2:instance",
            "resourceArns": [
                "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"
            ],
            "selectionMode": "ALL"
        }
    },
```

```
    "actions": {
        "reboot": {
            "actionId": "aws:ec2:reboot-instances",
            "description": "reboot",
            "parameters": {},
            "targets": {
                "Instances": "Instances-Target-1"
            }
        }
    },
    "roleArn": "arn:aws:iam::123456789012:role/myRole"
}
```

Sortie :

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "experimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "description": "reboot",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "aws:cloudwatch:alarm",
                "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
```

```
            }
        ],
        "creationTime": 1616434850.659,
        "lastUpdateTime": 1616434850.659,
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Création d'un modèle d'expérience dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous CreateExperimentTemplateà la section Référence des AWS CLI commandes.

`delete-experiment-template`

L'exemple de code suivant montre comment utiliser`delete-experiment-template`.

AWS CLI

Pour supprimer un modèle d'expérience

L'`delete-experiment-template`exemple suivant supprime le modèle d'expérience spécifié.

```
aws fis delete-experiment-template \
    --id ABCDE1fgHIJkLmNop
```

Sortie :

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "myExperimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
```

```
            }
        },
        "actions": {
            "testaction": {
                "actionId": "aws:ec2:stop-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616017191.124,
        "lastUpdateTime": 1616017859.607,
        "roleArn": "arn:aws:iam::123456789012:role/FISRole"
    }
}
```

Pour plus d'informations, voir Supprimer un modèle d'expérience dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous DeleteExperimentTemplateà la section Référence des AWS CLI commandes.

`get-action`

L'exemple de code suivant montre comment utiliser`get-action`.

AWS CLI

Pour obtenir les détails des actions

L'`get-action`exemple suivant permet d'obtenir les détails de l'action spécifiée.

```
aws fis get-action \
    --id aws:ec2:stop-instances
```

Sortie :

```
{
    "action": {
        "id": "aws:ec2:stop-instances",
        "description": "Stop the specified EC2 instances.",
        "parameters": {
            "startInstancesAfterDuration": {
                "description": "The time to wait before restarting the instances
 (ISO 8601 duration).",
                "required": false
            }
        },
        "targets": {
            "Instances": {
                "resourceType": "aws:ec2:instance"
            }
        },
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Actions du guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous GetActionà la section Référence des AWS CLI commandes.

`get-experiment-template`

L'exemple de code suivant montre comment utiliser`get-experiment-template`.

AWS CLI

Pour obtenir les détails du modèle d'expérience

L'`get-experiment-template`exemple suivant permet d'obtenir les détails du modèle d'expérience spécifié.

```
aws fis get-experiment-template \
    --id ABCDE1fgHIJkLmNop
```

Sortie :

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "myExperimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "testaction": {
                "actionId": "aws:ec2:stop-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616017191.124,
        "lastUpdateTime": 1616017331.51,
        "roleArn": "arn:aws:iam::123456789012:role/FISRole",
        "tags": {
        "key: "value"
        }
    }
}
```

Pour plus d'informations, consultez la section Modèles d'expériences dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous GetExperimentTemplateà la section Référence des AWS CLI commandes.

`get-experiment`

L'exemple de code suivant montre comment utiliser`get-experiment`.

AWS CLI

Pour obtenir les détails de l'expérience

L'`get-experiment`exemple suivant permet d'obtenir les détails de l'expérience spécifiée.

```
aws fis get-experiment \
    --id ABC12DeFGhI3jKLMNOP
```

Sortie :

```
{
    "experiment": {
        "id": "ABC12DeFGhI3jKLMNOP",
        "experimentTemplateId": "ABCDE1fgHIJkLmNop",
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "state": {
            "status": "completed",
            "reason": "Experiment completed."
        },
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                },
                "state": {
                    "status": "completed",
                    "reason": "Action was completed."
```

```
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616432509.662,
        "startTime": 1616432509.962,
        "endTime": 1616432522.307,
        "tags": {}
    }
}
```

Pour plus d'informations, voir Experiments for AWS FIS dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous GetExperimentà la section Référence des AWS CLI commandes.

`list-actions`

L'exemple de code suivant montre comment utiliser`list-actions`.

AWS CLI

Pour répertorier les actions

L'`list-actions`exemple suivant répertorie les actions disponibles.

```
aws fis list-actions
```

Sortie :

```
{
    "actions": [
        {
            "id": "aws:ec2:reboot-instances",
            "description": "Reboot the specified EC2 instances.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
```

```
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ec2:stop-instances",
            "description": "Stop the specified EC2 instances.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ec2:terminate-instances",
            "description": "Terminate the specified EC2 instances.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ecs:drain-container-instances",
            "description": "Drain percentage of underlying EC2 instances on an ECS
cluster.",
            "targets": {
                "Clusters": {
                    "resourceType": "aws:ecs:cluster"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:eks:terminate-nodegroup-instances",
            "description": "Terminates a percentage of the underlying EC2 instances
in an EKS cluster.",
            "targets": {
                "Nodegroups": {
                    "resourceType": "aws:eks:nodegroup"
                }
            },
```

```
            "tags": {}
        },
        {
            "id": "aws:fis:inject-api-internal-error",
            "description": "Cause an AWS service to return internal error responses
for specific callers and operations.",
            "targets": {
                "Roles": {
                    "resourceType": "aws:iam:role"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:fis:inject-api-throttle-error",
            "description": "Cause an AWS service to return throttled responses for
specific callers and operations.",
            "targets": {
                "Roles": {
                    "resourceType": "aws:iam:role"
                }
            },
            "tags": {}
        },
        {
        "id": "aws:fis:inject-api-unavailable-error",
            "description": "Cause an AWS service to return unavailable error
responses for specific callers and operations.",
            "targets": {
                "Roles": {
                    "resourceType": "aws:iam:role"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:fis:wait",
            "description": "Wait for the specified duration. Stop condition
monitoring will continue during this time.",
            "tags": {}
        },
        {
            "id": "aws:rds:failover-db-cluster",
            "description": "Failover a DB Cluster to one of the replicas.",
```

```
            "targets": {
                "Clusters": {
                    "resourceType": "aws:rds:cluster"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:rds:reboot-db-instances",
            "description": "Reboot the specified DB instances.",
            "targets": {
                "DBInstances": {
                    "resourceType": "aws:rds:db"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ssm:send-command",
            "description": "Run the specified SSM document.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
                }
            },
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Actions du guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous ListActionsà la section Référence des AWS CLI commandes.

`list-experiment-templates`

L'exemple de code suivant montre comment utiliser`list-experiment-templates`.

AWS CLI

Pour répertorier les modèles d'expériences

L'`list-experiment-templates`exemple suivant répertorie les modèles de test de votre AWS compte.

```
aws fis list-experiment-templates
```

Sortie :

```
{
    "experimentTemplates": [
        {
            "id": "ABCDE1fgHIJkLmNop",
            "description": "myExperimentTemplate",
            "creationTime": 1616017191.124,
            "lastUpdateTime": 1616017191.124,
            "tags": {
                "key": "value"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Modèles d'expériences dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous ListExperimentTemplatesà la section Référence des AWS CLI commandes.

`list-experiments`

L'exemple de code suivant montre comment utiliser`list-experiments`.

AWS CLI

Pour répertorier les expériences

L'`list-experiments`exemple suivant répertorie les expériences de votre AWS compte.

```
aws fis list-experiments
```

Sortie :

```
{
    "experiments": [
        {
            "id": "ABCdeF1GHiJkLM23NO",
            "experimentTemplateId": "ABCDE1fgHIJkLmNop",
            "state": {
                "status": "running",
                "reason": "Experiment is running."
            },
            "creationTime": 1616017341.197,
            "tags": {
            "key": "value"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Experiments in the AWS Fault Injection Simulator User Guide.

- Pour plus de détails sur l'API, reportez-vous ListExperimentsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises de la ressource spécifiée.

```
aws fis list-tags-for-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Sortie :

```
{
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
```

```
}
```

Pour plus d'informations, consultez la section Marquer vos ressources AWS FIS dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-experiment`

L'exemple de code suivant montre comment utiliser`start-experiment`.

AWS CLI

Pour démarrer une expérience

L'`start-experiment`exemple suivant démarre l'expérience spécifiée.

```
aws fis start-experiment \
    --experiment-template-id ABCDE1fgHIJkLmNop
```

Sortie :

```
{
    "experiment": {
        "id": "ABC12DeFGhI3jKLMNOP",
        "experimentTemplateId": "ABCDE1fgHIJkLmNop",
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "state": {
            "status": "initiating",
            "reason": "Experiment is initiating."
        },
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
```

```
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                },
                "state": {
                    "status": "pending",
                    "reason": "Initial state"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616432464.025,
        "startTime": 1616432464.374,
        "tags": {}
    }
}
```

Pour plus d'informations, voir Experiments for AWS FIS dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous StartExperimentà la section Référence des AWS CLI commandes.

`stop-experiment`

L'exemple de code suivant montre comment utiliser`stop-experiment`.

AWS CLI

Pour arrêter une expérience

L'`stop-experiment`exemple suivant arrête l'exécution de l'expérience spécifiée.

```
aws fis stop-experiment \
    --id ABC12DeFGhI3jKLMNOP
```

Sortie :


```
{
    "experiment": {
        "id": "ABC12DeFGhI3jKLMNOP",
        "experimentTemplateId": "ABCDE1fgHIJkLmNop",
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "state": {
            "status": "stopping",
            "reason": "Stopping Experiment."
        },
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                },
                "startAfter": [
                    "wait"
                ],
                "state": {
                    "status": "pending",
                    "reason": "Initial state."
                }
            },
            "wait": {
                "actionId": "aws:fis:wait",
                "parameters": {
                    "duration": "PT5M"
                },
                "state": {
                    "status": "running",
                    "reason": ""
```

```
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616432680.927,
        "startTime": 1616432681.177,
        "tags": {}
    }
}
```

Pour plus d'informations, voir Experiments for AWS FIS dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous StopExperimentà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant balise la ressource spécifiée.

```
aws fis tag-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
 \
    --tags key1=value1,key2=value2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquer vos ressources AWS FIS dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour annuler le balisage d'une ressource

L'`untag-resource`exemple suivant supprime les balises de la ressource spécifiée.

```
aws fis untag-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquer vos ressources AWS FIS dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-experiment-template`

L'exemple de code suivant montre comment utiliser`update-experiment-template`.

AWS CLI

Pour mettre à jour un modèle d'expérience

L'`update-experiment-template`exemple suivant met à jour la description du modèle d'expérience spécifié.

```
aws fis update-experiment-template \
    --id ABCDE1fgHIJkLmNop \
    ---description myExperimentTemplate
```

Sortie :

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "myExperimentTemplate",
        "targets": {
```

```
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "testaction": {
                "actionId": "aws:ec2:stop-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616017191.124,
        "lastUpdateTime": 1616017859.607,
        "roleArn": "arn:aws:iam::123456789012:role/FISRole",
        "tags": {
            "key": "value"
        }
    }
}
```

Pour plus d'informations, voir Mettre à jour un modèle d'expérience dans le guide de l'utilisateur du simulateur d'injection de AWS défauts.

- Pour plus de détails sur l'API, reportez-vous UpdateExperimentTemplateà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Amazon GameLift Servers AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide des GameLift serveurs Amazon.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-build`

L'exemple de code suivant montre comment utiliser`create-build`.

AWS CLI

Exemple1 : pour créer une version de jeu à partir de fichiers d'un compartiment S3

L'`create-build`exemple suivant crée une ressource de compilation de jeu personnalisée. Il utilise des fichiers compressés qui sont stockés dans un emplacement S3 d'un AWS compte que vous contrôlez. Cet exemple suppose que vous avez déjà créé un rôle IAM qui GameLift autorise Amazon à accéder à l'emplacement S3. Étant donné que la demande ne spécifie aucun système d'exploitation, la nouvelle ressource de compilation est par défaut WINDOWS_2012.

```
aws gamelift create-build \
    --storage-location file://storage-loc.json \
    --name MegaFrogRaceServer.NA \
    --build-version 12345.678
```

Contenu de `storage-loc.json` :

```
{
    "Bucket":"MegaFrogRaceServer_NA_build_files"
    "Key":"MegaFrogRaceServer_build_123.zip"
    "RoleArn":"arn:aws:iam::123456789012:role/gamelift"
}
```

Sortie :

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA",
        "OperatingSystem": "WINDOWS_2012",
        "SizeOnDisk": 479303,
        "Status": "INITIALIZED",
        "Version": "12345.678"
    },
    "StorageLocation": {
        "Bucket": "MegaFrogRaceServer_NA_build_files",
        "Key": "MegaFrogRaceServer_build_123.zip"
    }
}
```

Exemple2 : Pour créer une ressource de compilation de jeu permettant de télécharger manuellement des fichiers vers GameLift

L'`create-build`exemple suivant crée une nouvelle ressource de construction. Il dispose également d'un emplacement de stockage et d'informations d'identification temporaires qui vous permettent de télécharger manuellement votre build de GameLift jeu sur Amazon S3. Une fois que vous avez chargé votre build avec succès, le GameLift service valide le build et met à jour le statut du nouveau build.

```
aws gamelift create-build \
    --name MegaFrogRaceServer.NA \
    --build-version 12345.678 \
    --operating-system AMAZON_LINUX
```

Sortie :

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA",
        "OperatingSystem": "AMAZON_LINUX",
```

```
        "SizeOnDisk": 0,
        "Status": "INITIALIZED",
        "Version": "12345.678"
    },
    "StorageLocation": {
        "Bucket": "gamelift-builds-us-west-2",
        "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "UploadCredentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="
    }
}
```

Pour plus d'informations, consultez la section Upload a custom server build to GameLift dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateBuildà la section Référence des AWS CLI commandes.

`create-fleet`

L'exemple de code suivant montre comment utiliser`create-fleet`.

AWS CLI

Exemple 1 : pour créer un parc Linux de base

L'`create-fleet`exemple suivant crée un parc d'instances Linux à la demande configuré de manière minimale pour héberger une version de serveur personnalisée. Vous pouvez terminer la configuration en utilisant`update-fleet`.

```
aws gamelift create-fleet \
    --name MegaFrogRaceServer.NA.v2 \
    --description 'Hosts for v2 North America' \
    --build-id build-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
    --certificate-configuration 'CertificateType=GENERATED' \
    --ec2-instance-type c4.large \
    --fleet-type ON_DEMAND \
    --runtime-configuration 'ServerProcesses=[{LaunchPath=/local/game/release-na/
MegaFrogRace_Server.exe,ConcurrentExecutions=1}]'
```

Sortie :

```
{
    "FleetAttributes": {
        "BuildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        },
        "CreationTime": 1496365885.44,
        "Description": "Hosts for v2 North America",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetType": "ON_DEMAND",
        "InstanceType": "c4.large",
        "MetricGroups": ["default"],
        "Name": "MegaFrogRace.NA.v2",
        "NewGameSessionProtectionPolicy": "NoProtection",
        "OperatingSystem": "AMAZON_LINUX",
        "ServerLaunchPath": "/local/game/release-na/MegaFrogRace_Server.exe",
        "Status": "NEW"
    }
}
```

Exemple 2 : pour créer un parc Windows de base

L'`create-fleet`exemple suivant crée un parc d'instances Windows ponctuelles configuré de manière minimale pour héberger une version de serveur personnalisée. Vous pouvez terminer la configuration en utilisant`update-fleet`.

```
aws gamelift create-fleet \
    --name MegaFrogRace.NA.v2 \
    --description 'Hosts for v2 North America' \
    --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff  \
    --certificate-configuration 'CertificateType=GENERATED' \
    --ec2-instance-type c4.large \
    --fleet-type SPOT \
    --runtime-configuration 'ServerProcesses=[{LaunchPath=C:\game
\Bin64.Release.Dedicated\MegaFrogRace_Server.exe,ConcurrentExecutions=1}]'
```

Sortie :

```
{
    "FleetAttributes": {
        "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        },
        "CreationTime": 1496365885.44,
        "Description": "Hosts for v2 North America",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetType": "SPOT",
        "InstanceType": "c4.large",
        "MetricGroups": ["default"],
        "Name": "MegaFrogRace.NA.v2",
        "NewGameSessionProtectionPolicy": "NoProtection",
        "OperatingSystem": "WINDOWS_2012",
        "ServerLaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
        "Status": "NEW"
    }
}
```

Exemple 3 : pour créer une flotte entièrement configurée

L'`create-fleet`exemple suivant crée un parc d'instances Spot Windows pour une construction de serveur personnalisée, avec les paramètres de configuration les plus couramment utilisés fournis.

```
aws gamelift create-fleet \
    --name MegaFrogRace.NA.v2 \
    --description 'Hosts for v2 North America' \
    --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff \
    --certificate-configuration 'CertificateType=GENERATED' \
    --ec2-instance-type c4.large \
    --ec2-inbound-permissions
 'FromPort=33435,ToPort=33435,IpRange=10.24.34.0/23,Protocol=UDP' \
    --fleet-type SPOT \
    --new-game-session-protection-policy FullProtection \
    --runtime-configuration file://runtime-config.json \
    --metric-groups default \
    --instance-role-arn 'arn:aws:iam::444455556666:role/GameLiftS3Access'
```

Contenu de `runtime-config.json` :

```
GameSessionActivationTimeoutSeconds=300,
 MaxConcurrentGameSessionActivations=2,
 ServerProcesses=[
   {LaunchPath=C:\game\Bin64.Release.Dedicated\MegaFrogRace_Server.exe,Parameters=-
debug,ConcurrentExecutions=1},
   {LaunchPath=C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe,ConcurrentExecutions=1}]
```

Sortie :

```
{
    "FleetAttributes": {
        "InstanceRoleArn": "arn:aws:iam::444455556666:role/GameLiftS3Access",
        "Status": "NEW",
        "InstanceType": "c4.large",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "Description": "Hosts for v2 North America",
        "FleetType": "SPOT",
        "OperatingSystem": "WINDOWS_2012",
        "Name": "MegaFrogRace.NA.v2",
        "CreationTime": 1569309011.11,
        "MetricGroups": [
            "default"
        ],
        "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
        "ServerLaunchParameters": "abc",
        "ServerLaunchPath": "C:\\game\\Bin64.Release.Dedicated\
\MegaFrogRace_Server.exe",
        "NewGameSessionProtectionPolicy": "FullProtection",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        }
    }
}
```

Exemple 4 : pour créer un parc de serveurs en temps réel

L'`create-fleet`exemple suivant crée un parc d'instances Spot avec un script de configuration en temps réel qui a été chargé sur Amazon GameLift. Tous les serveurs en temps réel sont

déployés sur des machines Linux. Pour les besoins de cet exemple, supposons que le script en temps réel chargé inclut plusieurs fichiers de script, la `Init()` fonction située dans le fichier de script étant appeléeMainScript.js. Comme indiqué, ce fichier est identifié comme le script de lancement dans la configuration d'exécution.

```
aws gamelift create-fleet \
    --name MegaFrogRace.NA.realtime \
    --description 'Mega Frog Race Realtime fleet' \
    --script-id script-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
    --ec2-instance-type c4.large \
    --fleet-type SPOT \
    --certificate-configuration 'CertificateType=GENERATED' --runtime-configuration
 'ServerProcesses=[{LaunchPath=/local/game/MainScript.js,Parameters=+map
 Winter444,ConcurrentExecutions=5}]'
```

Sortie :

```
{
    "FleetAttributes": {
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "Status": "NEW",
        "CreationTime": 1569310745.212,
        "InstanceType": "c4.large",
        "NewGameSessionProtectionPolicy": "NoProtection",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        },
        "Name": "MegaFrogRace.NA.realtime",
        "ScriptId": "script-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetType": "SPOT",
        "MetricGroups": [
            "default"
        ],
        "Description": "Mega Frog Race Realtime fleet",
        "OperatingSystem": "AMAZON_LINUX"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFleetà la section Référence des AWS CLI commandes.

`create-game-session-queue`

L'exemple de code suivant montre comment utiliser`create-game-session-queue`.

AWS CLI

Exemple1 : Pour configurer une file d'attente ordonnée pour les sessions de jeu

L'`create-game-session-queue`exemple suivant crée une nouvelle file d'attente de session de jeu avec des destinations dans deux régions. Il configure également la file d'attente de manière à ce que les demandes de session de jeu expirent après 10 minutes d'attente pour être placées. Aucune politique de latence n'étant définie, GameLift tente de placer toutes les sessions de jeu avec la première destination répertoriée.

```
aws gamelift create-game-session-queue \
    --name MegaFrogRaceServer-NA \
    --destinations file://destinations.json \
    --timeout-in-seconds 600
```

Contenu de `destinations.json` :

```
{
    "Destinations": [
        {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },
        {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }
    ]
}
```

Sortie :

```
{
    "GameSessionQueues": [
        {
            "Name": "MegaFrogRaceServer-NA",
            "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:123456789012:gamesessionqueue/MegaFrogRaceServer-NA",
            "TimeoutInSeconds": 600,
            "Destinations": [
                {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},
```

```
                {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}
            ]
        }
    ]
}
```

Exemple 2 : pour configurer une file d'attente de session de jeu avec des politiques de latence pour les joueurs

L'`create-game-session-queue`exemple suivant crée une nouvelle file d'attente de session de jeu avec des politiques de latence pour deux joueurs. La première règle définit un plafond de latence de 100 ms qui est appliqué pendant la première minute d'une tentative de placement de session de jeu. La deuxième politique augmente le plafond de latence à 200 ms jusqu'à ce que la demande de placement expire au bout de 3 minutes.

```
aws gamelift create-game-session-queue \
    --name MegaFrogRaceServer-NA \
    --destinations file://destinations.json \
    --player-latency-policies file://latency-policies.json \
    --timeout-in-seconds 180
```

Contenu de `destinations.json` :

```
{
    "Destinations": [
        { "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },
        { "DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }
    ]
}
```

Contenu de `latency-policies.json` :

```
{
    "PlayerLatencyPolicies": [
        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
 60}
```

```
    ]
}
```

Sortie :

```
{
    "GameSessionQueue": {
        "Name": "MegaFrogRaceServer-NA",
        "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:111122223333:gamesessionqueue/MegaFrogRaceServer-NA",
        "TimeoutInSeconds": 600,
        "PlayerLatencyPolicies": [
            {
                "MaximumIndividualPlayerLatencyMilliseconds": 100,
                "PolicyDurationSeconds": 60
            },
            {
                "MaximumIndividualPlayerLatencyMilliseconds": 200
            }
        ]
        "Destinations": [
            {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},
            {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}
        ],
    }
}
```

Pour plus d'informations, consultez la section Créer une file d'attente dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateGameSessionQueueà la section Référence des AWS CLI commandes.

`delete-build`

L'exemple de code suivant montre comment utiliser`delete-build`.

AWS CLI

Pour supprimer une version de jeu personnalisée

L'`delete-build`exemple suivant supprime un build de votre GameLift compte Amazon. Une fois le build supprimé, vous ne pouvez pas l'utiliser pour créer de nouvelles flottes. Cette opération ne peut pas être annulée.

```
aws gamelift delete-build \
   --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBuildà la section Référence des AWS CLI commandes.

`delete-fleet`

L'exemple de code suivant montre comment utiliser`delete-fleet`.

AWS CLI

Pour supprimer un parc qui n'est plus utilisé

L'`delete-fleet`exemple suivant supprime une flotte qui a été réduite à zéro instance. Si la capacité du parc est supérieure à zéro, la demande échoue avec une erreur HTTP 400.

```
aws gamelift delete-fleet \
   --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gérer les GameLift flottes dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteFleetà la section Référence des AWS CLI commandes.

`delete-game-session-queue`

L'exemple de code suivant montre comment utiliser`delete-game-session-queue`.

AWS CLI

Pour supprimer une file d'attente de session de jeu

L'`delete-game-session-queue`exemple suivant supprime une file d'attente de session de jeu spécifiée.

```
aws gamelift delete-game-session-queue \
    --name MegaFrogRace-NA
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteGameSessionQueueà la section Référence des AWS CLI commandes.

`describe-build`

L'exemple de code suivant montre comment utiliser`describe-build`.

AWS CLI

Pour obtenir des informations sur une version de jeu personnalisée

L'`describe-build`exemple suivant récupère les propriétés d'une ressource de compilation d'un serveur de jeu.

```
aws gamelift describe-build \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "My_Game_Server_Build_One",
        "OperatingSystem": "AMAZON_LINUX",
        "SizeOnDisk": 1304924,
        "Status": "READY",
        "Version": "12345.678"
    }
}
```

Pour plus d'informations, consultez la section Upload a custom server build to GameLift dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeBuildà la section Référence des AWS CLI commandes.

`describe-ec2-instance-limits`

L'exemple de code suivant montre comment utiliser`describe-ec2-instance-limits`.

AWS CLI

Pour récupérer les limites de service pour un type d' EC2 instance

L'`describe-ec2-instance-limits`exemple suivant affiche le nombre maximum d'instances autorisées et les instances actuellement utilisées pour le type d' EC2 instance spécifié dans la région actuelle. Le résultat indique que seules cinq des vingt instances autorisées sont utilisées.

```
aws gamelift describe-ec2-instance-limits \
    --ec2-instance-type m5.large
```

Sortie :

```
{
    "EC2InstanceLimits": [
        {
            "EC2InstanceType": ""m5.large",
            "CurrentInstances": 5,
            "InstanceLimit": 20
        }
    ]
}
```

Pour plus d'informations, consultez Choose Computing Resources dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous à la section DescribeEc2 InstanceLimits de AWS CLI la section Référence des commandes.

`describe-fleet-attributes`

L'exemple de code suivant montre comment utiliser`describe-fleet-attributes`.

AWS CLI

Exemple1 : Pour afficher les attributs d'une liste de flottes

L'`describe-fleet-attributes`exemple suivant récupère les attributs de flotte pour deux flottes spécifiées. Comme indiqué, les flottes demandées sont déployées avec la même version, une pour les instances à la demande et une pour les instances Spot, avec quelques différences de configuration mineures.

```
aws gamelift describe-fleet-attributes \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Sortie :

```
{
    "FleetAttributes": [
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "FleetType": "ON_DEMAND",
            "InstanceType": "c4.large",
            "Description": "On-demand hosts for v2 North America",
            "Name": "MegaFrogRaceServer.NA.v2-od",
            "CreationTime": 1568836191.995,
            "Status": "ACTIVE",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
            "ServerLaunchParameters": "+gamelift_start_server",
            "NewGameSessionProtectionPolicy": "NoProtection",
            "OperatingSystem": "WINDOWS_2012",
            "MetricGroups": [
                "default"
            ],
            "CertificateConfiguration": {
                "CertificateType": "DISABLED"
            }
        },
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
            "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "FleetType": "SPOT",
            "InstanceType": "c4.large",
            "Description": "On-demand hosts for v2 North America",
            "Name": "MegaFrogRaceServer.NA.v2-spot",
            "CreationTime": 1568838275.379,
            "Status": "ACTIVATING",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
            "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
            "NewGameSessionProtectionPolicy": "NoProtection",
            "OperatingSystem": "WINDOWS_2012",
                "MetricGroups": [
                "default"
            ],
            "CertificateConfiguration": {
                "CertificateType": "GENERATED"
            }
        }
    ]
}
```

Exemple 2 : Pour demander des attributs pour toutes les flottes

Ce qui suit `describe-fleet-attributes` renvoie les attributs de flotte pour toutes les flottes, quel que soit leur statut. Cet exemple illustre l'utilisation de paramètres de pagination pour renvoyer une flotte à la fois.

```
aws gamelift describe-fleet-attributes \
    --limit 1
```

Sortie :

```
{
    "FleetAttributes": [
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "FleetType": "SPOT",
            "InstanceType": "c4.large",
```

```
            "Description": "On-demand hosts for v2 North America",
            "Name": "MegaFrogRaceServer.NA.v2-spot",
            "CreationTime": 1568838275.379,
            "Status": "ACTIVATING",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
            "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
            "NewGameSessionProtectionPolicy": "NoProtection",
            "OperatingSystem": "WINDOWS_2012",
            "MetricGroups": [
                "default"
            ],
            "CertificateConfiguration": {
                "CertificateType": "GENERATED"
            }
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

La sortie inclut une `NextToken` valeur que vous pouvez utiliser lorsque vous appelez la commande une deuxième fois. Transmettez la valeur au `--next-token` paramètre pour spécifier où récupérer la sortie. La commande suivante renvoie le deuxième résultat dans la sortie.

```
aws gamelift describe-fleet-attributes \
    --limit 1 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Répétez l'opération jusqu'à ce que la réponse n'inclue aucune `NextToken` valeur.

Pour plus d'informations, consultez la section Configuration GameLift des flottes dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetAttributesà la section Référence des AWS CLI commandes.

`describe-fleet-capacity`

L'exemple de code suivant montre comment utiliser`describe-fleet-capacity`.

AWS CLI

Pour consulter l'état de capacité d'une liste de flottes

L'`describe-fleet-capacity`exemple suivant permet de récupérer la capacité actuelle de deux flottes spécifiées.

```
aws gamelift describe-fleet-capacity \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Sortie :

```
{
    "FleetCapacity": [
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "InstanceType": "c5.large",
            "InstanceCounts": {
                "DESIRED": 10,
                "MINIMUM": 1,
                "MAXIMUM": 20,
                "PENDING": 0,
                "ACTIVE": 10,
                "IDLE": 3,
                "TERMINATING": 0
            }
        },
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "InstanceType": "c5.large",
            "InstanceCounts": {
                "DESIRED": 13,
                "MINIMUM": 1,
                "MAXIMUM": 20,
                "PENDING": 0,
                "ACTIVE": 15,
                "IDLE": 2,
                "TERMINATING": 2
            }
        }

    ]
```

```
}
```

Pour plus d'informations, consultez la section GameLift Metrics for Fleets dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetCapacityà la section Référence des AWS CLI commandes.

`describe-fleet-events`

L'exemple de code suivant montre comment utiliser`describe-fleet-events`.

AWS CLI

Pour demander des événements pendant une période spécifiée

L'`describe-fleet-events`exemple suivant affiche les détails de tous les événements liés à la flotte survenus au cours de la période spécifiée.

```
aws gamelift describe-fleet-events \
    --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111 \
    --start-time 1579647600 \
    --end-time 1579649400 \
    --limit 5
```

Sortie :

```
{
    "Events": [
        {
            "EventId": "a37b6892-5d07-4d3b-8b47-80244ecf66b9",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_ACTIVE",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to ACTIVE",
            "EventTime": 1579649342.191
        },
        {
            "EventId": "67da4ec9-92a3-4d95-886a-5d6772c24063",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_ACTIVATING",
```

```
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to ACTIVATING",
            "EventTime": 1579649321.427
        },
        {
            "EventId": "23813a46-a9e6-4a53-8847-f12e6a8381ac",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_BUILDING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to BUILDING",
            "EventTime": 1579649321.243
        },
        {
            "EventId": "3bf217d0-1d44-42f9-9202-433ed475d2e8",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_VALIDATING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to VALIDATING",
            "EventTime": 1579649197.449
        },
        {
            "EventId": "2ecd0130-5986-44eb-99a7-62df27741084",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND",
            "Message": "Failed to find a valid path",
            "EventTime": 1569319075.839,
            "PreSignedLogUrl": "https://gamelift-event-logs-prod-
us-west-2.s3.us-west-2.amazonaws.com/logs/fleet-83422059-8329-42a2-
a4d6-c4444386a6f8/events/2ecd0130-5986-44eb-99a7-62df27741084/
FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEB8aCXVzLXdlc3QtMiJHMEUCIHV5K%2FLPx8h310D
%2FAvx0%2FZxsDy5XA3cJOwPdu3T0eBa%2FAiEA1yovokcZYy%2FV4CWW6l26aFyiSHO
%2Bxz%2FBMAhEHYHMQNcqkQMImP%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F
%2FARAAGgw3NDEwNjE1OTIxNzEiDI8rsZtzLzlwEDQhXSrlAtl5Ae
%2Fgo6FCIzqXPbXfBOnSvFYqeDlriZarEpKqKrUt8mXQv9iqHResqCph9AKo49lwgSYTT2QoSxnrD7%2FUgv
%2BZm2pVuczvuKtUA0fcx6s0GxpjIAzdIE%2F5P%2FB7B9M%2BVZ
%2F9KF82hbJi0HTE6Y7BjKsEgFCvk4UXILhfjtan9iQl8%2F21ZTurAcJbm7Y5tuLF9SWSK3%2BEa7VXOcCK4D4O1sMj
%2FIaXoHkNvg0RVTa0hIqdvpaDQlsSBNdqTXbjHTu6fETE9Y9Ky%2BiJK5KiUG
%2F59GjCpDcvS1FqKeLUEmKT7wysGmvjMc2n%2Fr
%2F9VxQfte7w9srXwlLAQuwhiXAAyI5ICMZ5JvzjzQwTqD4CHTVKUUDwL
%2BRZzbuuqkJObZml02CkRGp%2B74RTAzLbWptVqZTIfzctiCTmWxb
%2FmKyELRYsVLrwNJ%2BGJ7%2BCrN0RC%2FjlgfLYIZyeAqjPgAu5HjgX
%2BM7jCo9M7wBTrnAXKOFQuf9dvA84SuwXOJFp17LYGjrHMKv0qC3GfbTMrZ6kzeNV9awKCpXB2Gnx9z2KvIlJdqirWV
%2F9C6%2B4jIZPME3jXmZcEHqqw5uvAVF7aeIavtUZU8pxpDIWT0YE4p3Kriy2AA7ziCRKtVfjV839InyLk8LUjsioWK
```

```
%2BYUq8%2FDTlLxqj1S%2FiO4TI0Wo7ilAo%2FKKWWF4guuNDexj8EOOynSp1yImB
%2BZf2Fua3O44W4eEXAMPLE33333&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170621T231808Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20170621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

Pour plus d'informations, consultez la section Debug GameLift Fleet Issues dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetEventsà la section Référence des AWS CLI commandes.

`describe-fleet-port-settings`

L'exemple de code suivant montre comment utiliser`describe-fleet-port-settings`.

AWS CLI

Pour consulter les autorisations de connexion entrante pour une flotte

L'`describe-fleet-port-settings`exemple suivant récupère les paramètres de connexion pour un parc spécifique.

```
aws gamelift describe-fleet-port-settings \
    --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111
```

Sortie :

```
{
    "InboundPermissions": [
        {
            "FromPort": 33400,
            "ToPort": 33500,
            "IpRange": "0.0.0.0/0",
            "Protocol": "UDP"
        },
        {
```

```
            "FromPort": 1900,
            "ToPort": 2000,
            "IpRange": "0.0.0.0/0",
            "Protocol": "TCP"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration GameLift des flottes dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetPortSettingsà la section Référence des AWS CLI commandes.

`describe-fleet-utilization`

L'exemple de code suivant montre comment utiliser`describe-fleet-utilization`.

AWS CLI

Exemple1 : Pour afficher les données d'utilisation d'une liste de flottes

L'`describe-fleet-utilization`exemple suivant permet de récupérer les informations d'utilisation actuelles pour un parc spécifique.

```
aws gamelift describe-fleet-utilization \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "FleetUtilization": [
        {
        "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ActiveServerProcessCount": 100,
        "ActiveGameSessionCount": 62,
        "CurrentPlayerSessionCount": 329,
        "MaximumPlayerSessionCount": 1000
        }
    ]
}
```

Exemple 2 : Pour demander des données d'utilisation pour toutes les flottes

Ce qui suit `describe-fleet-utilization` renvoie les données d'utilisation de la flotte pour toutes les flottes, quel que soit leur statut. Cet exemple utilise des paramètres de pagination pour renvoyer des données pour deux flottes à la fois.

```
aws gamelift describe-fleet-utilization \
    --limit 2
```

Sortie :

```
{
    "FleetUtilization": [
        {
            "FleetId": "fleet-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
            "ActiveServerProcessCount": 100,
            "ActiveGameSessionCount": 13,
            "CurrentPlayerSessionCount": 98,
            "MaximumPlayerSessionCount": 1000
        },
        {
            "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
            "ActiveServerProcessCount": 100,
            "ActiveGameSessionCount": 62,
            "CurrentPlayerSessionCount": 329,
            "MaximumPlayerSessionCount": 1000
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

Appelez la commande une deuxième fois, en transmettant la `NextToken` valeur comme argument au `--next-token` paramètre pour voir les deux résultats suivants.

```
aws gamelift describe-fleet-utilization \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Répétez l'opération jusqu'à ce que la réponse n'inclue plus de `NextToken` valeur dans la sortie.

Pour plus d'informations, consultez la section GameLift Metrics for Fleets dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeFleetUtilizationà la section Référence des AWS CLI commandes.

`describe-game-session-queues`

L'exemple de code suivant montre comment utiliser`describe-game-session-queues`.

AWS CLI

Pour afficher les files d'attente des sessions de jeu

L'`describe-game-session-queues`exemple suivant récupère les propriétés de deux files d'attente spécifiées.

```
aws gamelift describe-game-session-queues \
    --names MegaFrogRace-NA MegaFrogRace-EU
```

Sortie :

```
{
    "GameSessionQueues": [{
            "Destinations": [{
                    "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                {
                    "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
                }
            ],
            "Name": "MegaFrogRace-NA",
            "TimeoutInSeconds": 600,
            "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/
MegaFrogRace-NA",
            "PlayerLatencyPolicies": [{
                    "MaximumIndividualPlayerLatencyMilliseconds": 200
                },
                {
                    "MaximumIndividualPlayerLatencyMilliseconds": 100,
                    "PolicyDurationSeconds": 60
```

```
                }
            ],
            "FilterConfiguration": {
                "AllowedLocations": ["us-west-2", "ap-south-1", "us-east-1"]
            },
            "PriorityConfiguration": {
                "PriorityOrder": ["LOCATION", "FLEET_TYPE", "DESTINATION"],
                "LocationOrder": ["us-west-2", "ap-south-1", "us-east-1"]
            }
        },
        {
            "Destinations": [{
                "DestinationArn": "arn:aws:gamelift:eu-west-3::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
            }],
            "Name": "MegaFrogRace-EU",
            "TimeoutInSeconds": 600,
            "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/
MegaFrogRace-EU"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des files d'attente multirégionales dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeGameSessionQueuesà la section Référence des AWS CLI commandes.

`describe-runtime-configuration`

L'exemple de code suivant montre comment utiliser`describe-runtime-configuration`.

AWS CLI

Pour demander la configuration d'exécution d'une flotte

L'`describe-runtime-configuration`exemple suivant permet de récupérer des informations sur la configuration d'exécution actuelle pour un parc spécifique.

```
aws gamelift describe-runtime-configuration \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RuntimeConfiguration": {
        "ServerProcesses": [
            {
                "LaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
                "Parameters": "+gamelift_start_server",
                "ConcurrentExecutions": 3
            },
            {
                "LaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
                "Parameters": "+gamelift_start_server +debug",
                "ConcurrentExecutions": 1
            }
        ],
        "MaxConcurrentGameSessionActivations": 2147483647,
        "GameSessionActivationTimeoutSeconds": 300
    }
}
```

Pour plus d'informations, consultez la section Exécuter plusieurs processus sur un parc dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeRuntimeConfigurationà la section Référence des AWS CLI commandes.

`list-builds`

L'exemple de code suivant montre comment utiliser`list-builds`.

AWS CLI

Exemple1 : Pour obtenir une liste des versions de jeu personnalisées

L'`list-builds`exemple suivant récupère les propriétés de toutes les versions de serveurs de jeu de la région actuelle. L'exemple de demande montre comment utiliser les paramètres de pagination `Limit` et comment récupérer `NextToken` les résultats dans des ensembles séquentiels. La première commande récupère les deux premières versions. Comme il y en a plus de deux, la réponse inclut un `NextToken` pour indiquer que d'autres résultats sont disponibles.

```
aws gamelift list-builds \
    --limit 2
```

Sortie :

```
{
    "Builds": [
        {
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "CreationTime": 1495664528.723,
            "Name": "My_Game_Server_Build_One",
            "OperatingSystem": "WINDOWS_2012",
            "SizeOnDisk": 8567781,
            "Status": "READY",
            "Version": "12345.678"
        },
        {
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "CreationTime": 1495528748.555,
            "Name": "My_Game_Server_Build_Two",
            "OperatingSystem": "AMAZON_LINUX_2",
            "SizeOnDisk": 8567781,
            "Status": "FAILED",
            "Version": "23456.789"
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

Vous pouvez ensuite appeler à nouveau la commande avec le `--next-token` paramètre suivant pour voir les deux prochaines versions.

```
aws gamelift list-builds \
    --limit 2
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Répétez l'opération jusqu'à ce que la réponse n'inclue aucune `NextToken` valeur.

Exemple2 : Pour obtenir une liste des versions de jeu personnalisées en état d'échec

L'`list-builds`exemple suivant récupère les propriétés de toutes les versions de serveurs de jeu de la région actuelle qui ont actuellement le statut FAILED.

```
aws gamelift list-builds \
    --status FAILED
```

Sortie :

```
{
    "Builds": [
        {
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "CreationTime": 1495528748.555,
            "Name": "My_Game_Server_Build_Two",
            "OperatingSystem": "AMAZON_LINUX_2",
            "SizeOnDisk": 8567781,
            "Status": "FAILED",
            "Version": "23456.789"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListBuildsà la section Référence des AWS CLI commandes.

`list-fleets`

L'exemple de code suivant montre comment utiliser`list-fleets`.

AWS CLI

Exemple1 : Pour obtenir la liste de toutes les flottes d'une région

L'`list-fleets`exemple suivant montre la flotte IDs de toutes les flottes de la région actuelle. Cet exemple utilise des paramètres de pagination pour récupérer deux flottes IDs à la fois. La réponse inclut un `next-token` attribut qui indique qu'il y a d'autres résultats à récupérer.

```
aws gamelift list-fleets \
    --limit 2
```

Sortie :

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

Vous pouvez transmettre la `NextToken` valeur de la réponse précédente dans la commande suivante, comme indiqué ici pour obtenir les deux résultats suivants.

```
aws gamelift list-fleets \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRlZj
```

Exemple2 : Pour obtenir une liste de toutes les flottes d'une région avec une version ou un script spécifique

L'`list-builds`exemple suivant récupère les IDs flottes déployées avec la version de jeu spécifiée. Si vous travaillez avec des serveurs en temps réel, vous pouvez fournir un identifiant de script à la place d'un identifiant de build. Comme cet exemple ne spécifie pas le paramètre de limite, les résultats peuvent inclure jusqu'à 16 flottes IDs.

```
aws gamelift list-fleets \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListFleetsà la section Référence des AWS CLI commandes.

`request-upload-credentials`

L'exemple de code suivant montre comment utiliser`request-upload-credentials`.

AWS CLI

Pour actualiser les informations d'accès pour le téléchargement d'une version

L'`create-build`exemple suivant permet d'obtenir de nouvelles informations d'accès valides pour le téléchargement d'un fichier de GameLift build vers un emplacement Amazon S3. Les informations d'identification ont une durée de vie limitée. Vous obtenez l'ID de build à partir de la réponse à la `CreateBuild` demande d'origine.

```
aws gamelift request-upload-credentials \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "StorageLocation": {
        "Bucket": "gamelift-builds-us-west-2",
        "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "UploadCredentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="
    }
}
```

Pour plus d'informations, consultez la section Upload a custom server build to GameLift dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RequestUploadCredentialsà la section Référence des AWS CLI commandes.

`start-fleet-actions`

L'exemple de code suivant montre comment utiliser`start-fleet-actions`.

AWS CLI

Pour relancer l'activité de mise à l'échelle automatique de la flotte

L'`start-fleet-actions`exemple suivant reprend l'utilisation de toutes les politiques de dimensionnement définies pour le parc spécifié mais qui ont été arrêtées en stop-fleet-actions appelant`` ``. Après le démarrage, les politiques de dimensionnement commencent immédiatement à suivre leurs indicateurs respectifs.

```
aws gamelift start-fleet-actions \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --actions AUTO_SCALING
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous StartFleetActionsà la section Référence des AWS CLI commandes.

`stop-fleet-actions`

L'exemple de code suivant montre comment utiliser`stop-fleet-actions`.

AWS CLI

Pour arrêter l'activité de mise à l'échelle automatique d'une flotte

L'`stop-fleet-actions`exemple suivant arrête l'utilisation de toutes les politiques de dimensionnement définies pour le parc spécifié. Une fois les politiques suspendues, la capacité du parc reste au même niveau que le nombre d'instances actives, sauf si vous l'ajustez manuellement.

```
aws gamelift start-fleet-actions \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --actions AUTO_SCALING
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous StopFleetActionsà la section Référence des AWS CLI commandes.

`update-build`

L'exemple de code suivant montre comment utiliser`update-build`.

AWS CLI

Pour mettre à jour une version de jeu personnalisée

L'`update-build`exemple suivant modifie le nom et les informations de version associés à une ressource de génération spécifiée. L'objet de construction renvoyé vérifie que les modifications ont été effectuées avec succès.

```
aws gamelift update-build \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --name MegaFrogRaceServer.NA.east \
    --build-version 12345.east
```

Sortie :

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA.east",
        "OperatingSystem": "AMAZON_LINUX_2",
        "SizeOnDisk": 1304924,
        "Status": "READY",
        "Version": "12345.east"
    }
}
```

Pour plus d'informations, consultez Mettre à jour vos fichiers de compilation dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateBuildà la section Référence des AWS CLI commandes.

`update-game-session-queue`

L'exemple de code suivant montre comment utiliser`update-game-session-queue`.

AWS CLI

Pour mettre à jour la configuration d'une file d'attente de session de jeu

L'`update-game-session-queue`exemple suivant ajoute une nouvelle destination et met à jour les politiques de latence des joueurs pour une file d'attente de session de jeu existante.

```
aws gamelift update-game-session-queue \
    --name MegaFrogRace-NA \
    --destinations file://destinations.json \
    --player-latency-policies file://latency-policies.json
```

Contenu de `destinations.json` :

```
{
    "Destinations": [
        {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
        {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
        {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
    ]
}
```

Contenu de `latency-policies.json` :

```
{
    "PlayerLatencyPolicies": [
        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
 120},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
 120}
    ]
}
```

Sortie :

```
{
    "GameSessionQueue": {
        "Destinations": [
            {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
            {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
            {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
        ],
        "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:111122223333:gamesessionqueue/MegaFrogRace-NA",
        "Name": "MegaFrogRace-NA",
        "TimeoutInSeconds": 600,
        "PlayerLatencyPolicies": [
            {"MaximumIndividualPlayerLatencyMilliseconds": 200},
            {"MaximumIndividualPlayerLatencyMilliseconds": 150,
 "PolicyDurationSeconds": 120},
            {"MaximumIndividualPlayerLatencyMilliseconds": 100,
 "PolicyDurationSeconds": 120}
        ]
    }
}
```

Pour plus d'informations, consultez la section Utilisation des files d'attente multirégionales dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateGameSessionQueueà la section Référence des AWS CLI commandes.

`upload-build`

L'exemple de code suivant montre comment utiliser`upload-build`.

AWS CLI

Exemple1 : Pour télécharger une version de serveur de jeu Linux

L'`upload-build`exemple suivant télécharge les fichiers de build d'un serveur de jeu Linux depuis un répertoire de fichiers vers le GameLift service et crée une ressource de compilation.

```
aws gamelift upload-build \
```

```
    --name MegaFrogRaceServer.NA \
    --build-version 2.0.1 \
    --build-root ~/MegaFrogRace_Server/release-na \
    --operating-system AMAZON_LINUX_2
    --server-sdk-version 4.0.2
```

Sortie :

```
Uploading ~/MegaFrogRace_Server/release-na:  16.0 KiB / 74.6 KiB (21.45%)
Uploading ~/MegaFrogRace_Server/release-na:  32.0 KiB / 74.6 KiB (42.89%)
Uploading ~/MegaFrogRace_Server/release-na:  48.0 KiB / 74.6 KiB (64.34%)
Uploading ~/MegaFrogRace_Server/release-na:  64.0 KiB / 74.6 KiB (85.79%)
Uploading ~/MegaFrogRace_Server/release-na:  74.6 KiB / 74.6 KiB (100.00%)
Successfully uploaded ~/MegaFrogRace_Server/release-na to AWS GameLift
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Exemple2 : Pour télécharger une version de serveur de jeu Windows

L'`upload-build`exemple suivant télécharge les fichiers de build du serveur de jeu Windows depuis un répertoire vers le GameLift service et crée un enregistrement de build.

```
aws gamelift upload-build \
    --name MegaFrogRaceServer.NA \
    --build-version 2.0.1 \
    --build-root C:\MegaFrogRace_Server\release-na \
    --operating-system WINDOWS_2012
    --server-sdk-version 4.0.2
```

Sortie :

```
Uploading C:\MegaFrogRace_Server\release-na:  16.0 KiB / 74.6 KiB (21.45%)
Uploading C:\MegaFrogRace_Server\release-na:  32.0 KiB / 74.6 KiB (42.89%)
Uploading C:\MegaFrogRace_Server\release-na:  48.0 KiB / 74.6 KiB (64.34%)
Uploading C:\MegaFrogRace_Server\release-na:  64.0 KiB / 74.6 KiB (85.79%)
Uploading C:\MegaFrogRace_Server\release-na:  74.6 KiB / 74.6 KiB (100.00%)
Successfully uploaded C:\MegaFrogRace_Server\release-na to AWS GameLift
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Pour plus d'informations, consultez la section Upload a custom server build to GameLift dans le manuel Amazon GameLift Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UploadBuildà la section Référence des AWS CLI commandes.

Exemples de Global Accelerator utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l'accélérateur global AWS Command Line Interface with.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-custom-routing-endpoints`

L'exemple de code suivant montre comment utiliser`add-custom-routing-endpoints`.

AWS CLI

Pour ajouter un point de terminaison de sous-réseau VPC à un groupe de points de terminaison pour un accélérateur de routage personnalisé

L'`add-custom-routing-endpoints`exemple suivant ajoute un point de terminaison de sous-réseau VPC à un groupe de points de terminaison pour un accélérateur de routage personnalisé.

```
aws globalaccelerator add-custom-routing-endpoints \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd \
    --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

Sortie :

```
{
    "EndpointDescriptions": [
        {
            "EndpointId": "subnet-1234567890abcdef0"
        }
    ],

 "EndpointGroupArn":"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd"
}
```

Pour plus d'informations, consultez la section Points de terminaison du sous-réseau VPC pour les accélérateurs de routage personnalisés dans Global Accelerator dans le AWS guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir AddCustomRoutingEndpointsla section Référence des AWS CLI commandes.

`advertise-byoip-cidr`

L'exemple de code suivant montre comment utiliser`advertise-byoip-cidr`.

AWS CLI

Pour annoncer une plage d'adresses

L'`advertise-byoip-cidr`exemple suivant demande AWS de publier une plage d'adresses que vous avez configurée pour être utilisée avec vos AWS ressources.

```
aws globalaccelerator advertise-byoip-cidr \
    --cidr 198.51.100.0/24
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "198.51.100.0/24",
        "State": "PENDING_ADVERTISING"
    }
}
```

Pour plus d'informations, consultez la section Apportez votre propre adresse IP dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir AdvertiseByoipCidrla section Référence des AWS CLI commandes.

`allow-custom-routing-traffic`

L'exemple de code suivant montre comment utiliser`allow-custom-routing-traffic`.

AWS CLI

Pour autoriser le trafic vers des destinations d' EC2 instance Amazon spécifiques dans un sous-réseau VPC pour un accélérateur de routage personnalisé

L'`allow-custom-routing-traffic`exemple suivant indique que le trafic est autorisé vers certaines adresses IP et certains ports d' EC2 instance Amazon (destination) pour qu'un point de terminaison de sous-réseau VPC dans un accélérateur de routage personnalisé puisse recevoir du trafic.

```
aws globalaccelerator allow-custom-routing-traffic \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example \
    --endpoint-id subnet-abcd123example \
    --destination-addresses "172.31.200.6" "172.31.200.7" \
    --destination-ports 80 81
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Points de terminaison du sous-réseau VPC pour les accélérateurs de routage personnalisés dans Global Accelerator dans le AWS guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir AllowCustomRoutingTrafficla section Référence des AWS CLI commandes.

`create-accelerator`

L'exemple de code suivant montre comment utiliser`create-accelerator`.

AWS CLI

Pour créer un accélérateur

L'`create-accelerator`exemple suivant crée un accélérateur avec deux balises avec deux adresses IP statiques BYOIP. Vous devez spécifier la `US-West-2 (Oregon)` région pour créer ou mettre à jour un accélérateur.

```
aws globalaccelerator create-accelerator \
    --name ExampleAccelerator \
    --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \
    --ip-addresses 192.0.2.250 198.51.100.52
```

Sortie :

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847.0,
        "LastModifiedTime": 1542394847.0
    }
}
```

Pour plus d'informations, voir Accélérateurs dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir CreateAcceleratorla section Référence des AWS CLI commandes.

`create-custom-routing-accelerator`

L'exemple de code suivant montre comment utiliser`create-custom-routing-accelerator`.

AWS CLI

Pour créer un accélérateur de routage personnalisé

L'`create-custom-routing-accelerator`exemple suivant crée un accélérateur de routage personnalisé avec les balises `Name` et`Project`.

```
aws globalaccelerator create-custom-routing-accelerator \
    --name ExampleCustomRoutingAccelerator \
    --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \
    --ip-addresses 192.0.2.250 198.51.100.52
```

Sortie :

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleCustomRoutingAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847.0,
        "LastModifiedTime": 1542394847.0
```

```
    }
}
```

Pour plus d'informations, consultez la section Accélérateurs de routage personnalisés dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir CreateCustomRoutingAcceleratorla section Référence des AWS CLI commandes.

`create-custom-routing-endpoint-group`

L'exemple de code suivant montre comment utiliser`create-custom-routing-endpoint-group`.

AWS CLI

Pour créer un groupe de points de terminaison pour un accélérateur de routage personnalisé

L'`create-custom-routing-endpoint-group`exemple suivant crée un groupe de points de terminaison pour un accélérateur de routage personnalisé.

```
aws globalaccelerator create-custom-routing-endpoint-group \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --endpoint-group-region us-east-2 \
    --destination-configurations "FromPort=80,ToPort=81,Protocols=TCP,UDP"
```

Sortie :

```
{
    "EndpointGroup": {
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd",
        "EndpointGroupRegion": "us-east-2",
        "DestinationDescriptions": [
            {
                "FromPort": 80,
                "ToPort": 81,
                "Protocols": [
                    "TCP",
                    "UDP"
                ]
```

```
            }
        ],
        "EndpointDescriptions": []
    }
}
```

Pour plus d'informations, consultez la section Groupes de points de terminaison pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir CreateCustomRoutingEndpointGroupla section Référence des AWS CLI commandes.

`create-custom-routing-listener`

L'exemple de code suivant montre comment utiliser`create-custom-routing-listener`.

AWS CLI

Pour créer un écouteur pour un accélérateur de routage personnalisé

L'`create-custom-routing-listener`exemple suivant crée un écouteur avec une plage de ports comprise entre 5000 et 10000 pour un accélérateur de routage personnalisé.

```
aws globalaccelerator create-custom-routing-listener \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --port-ranges FromPort=5000,ToPort=10000
```

Sortie :

```
{
    "Listener": {
        "PortRange": [
            "FromPort": 5000,
            "ToPort": 10000
        ],
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz"
    }
}
```

Pour plus d'informations, consultez la section Écouteurs pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir CreateCustomRoutingListenerla section Référence des AWS CLI commandes.

`create-endpoint-group`

L'exemple de code suivant montre comment utiliser`create-endpoint-group`.

AWS CLI

Pour créer un groupe de points de terminaison

L'`create-endpoint-group`exemple suivant crée un groupe de points de terminaison avec un point de terminaison.

```
aws globalaccelerator create-endpoint-group \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --endpoint-group-region us-east-1 \
    --endpoint-configurations EndpointId=i-1234567890abcdef0,Weight=128
```

Sortie :

```
{
    "EndpointGroup": {
        "TrafficDialPercentage": 100.0,
        "EndpointDescriptions": [
            {
                "Weight": 128,
                "EndpointId": "i-1234567890abcdef0"
            }
        ],
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
        "EndpointGroupRegion": "us-east-1"
    }
}
```

Pour plus d'informations, consultez la section Groupes de terminaux dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir CreateEndpointGroupla section Référence des AWS CLI commandes.

`create-listener`

L'exemple de code suivant montre comment utiliser`create-listener`.

AWS CLI

Pour créer un écouteur

L'`create-listener`exemple suivant crée un écouteur avec deux ports.

```
aws globalaccelerator create-listener \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --port-ranges FromPort=80,ToPort=80 FromPort=81,ToPort=81 \
    --protocol TCP
```

Sortie :

```
{
    "Listener": {
        "PortRanges": [
            {
                "ToPort": 80,
                "FromPort": 80
            },
            {
                "ToPort": 81,
                "FromPort": 81
            }
        ],
        "ClientAffinity": "NONE",
        "Protocol": "TCP",
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz"
    }
```

```
}
```

Pour plus d'informations, voir Listeners in AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir CreateListenerla section Référence des AWS CLI commandes.

`deny-custom-routing-traffic`

L'exemple de code suivant montre comment utiliser`deny-custom-routing-traffic`.

AWS CLI

Pour spécifier une adresse de destination qui ne peut pas recevoir de trafic dans un accélérateur de routage personnalisé

L'`deny-custom-routing-traffic`exemple suivant indique l'adresse ou les adresses de destination d'un point de terminaison de sous-réseau qui ne peuvent pas recevoir de trafic pour un accélérateur de routage personnalisé. Pour spécifier plusieurs adresses de destination, séparez-les par un espace. Il n'y a pas de réponse en cas d' deny-custom-routing-trafficappel réussi.

```
aws globalaccelerator deny-custom-routing-traffic \
    --endpoint-group-
arn "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example" \
    --endpoint-id "subnet-abcd123example" \
    --destination-addresses "198.51.100.52"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Points de terminaison du sous-réseau VPC pour les accélérateurs de routage personnalisés dans Global Accelerator dans le AWS guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DenyCustomRoutingTrafficla section Référence des AWS CLI commandes.

`deprovision-byoip-cidr`

L'exemple de code suivant montre comment utiliser`deprovision-byoip-cidr`.

AWS CLI

Pour déprovisionner une plage d'adresses

L'`deprovision-byoip-cidr`exemple suivant publie la plage d'adresses spécifiée que vous avez configurée pour être utilisée avec vos AWS ressources.

```
aws globalaccelerator deprovision-byoip-cidr \
    --cidr "198.51.100.0/24"
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "198.51.100.0/24",
        "State": "PENDING_DEPROVISIONING"
    }
}
```

Pour plus d'informations, consultez la section Apportez votre propre adresse IP dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DeprovisionByoipCidrla section Référence des AWS CLI commandes.

`describe-accelerator-attributes`

L'exemple de code suivant montre comment utiliser`describe-accelerator-attributes`.

AWS CLI

Pour décrire les attributs d'un accélérateur

L'`describe-accelerator-attributes`exemple suivant récupère les détails de l'attribut d'un accélérateur.

```
aws globalaccelerator describe-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

Pour plus d'informations, voir Accélérateurs dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeAcceleratorAttributesla section Référence des AWS CLI commandes.

`describe-accelerator`

L'exemple de code suivant montre comment utiliser`describe-accelerator`.

AWS CLI

Pour décrire un accélérateur

L'`describe-accelerator`exemple suivant permet de récupérer les informations relatives à l'accélérateur spécifié.

```
aws globalaccelerator describe-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
```

```
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847,
        "LastModifiedTime": 1542395013
    }
}
```

Pour plus d'informations, voir Accélérateurs dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeAcceleratorla section Référence des AWS CLI commandes.

`describe-custom-routing-accelerator-attributes`

L'exemple de code suivant montre comment utiliser`describe-custom-routing-accelerator-attributes`.

AWS CLI

Pour décrire les attributs d'un accélérateur de routage personnalisé

L'`describe-custom-routing-accelerator-attributes`exemple suivant décrit les attributs d'un accélérateur de routage personnalisé.

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \
   --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "AcceleratorAttributes": {
    "FlowLogsEnabled": false
    }
```

```
}
```

Pour plus d'informations, consultez la section Accélérateurs de routage personnalisés dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeCustomRoutingAcceleratorAttributesla section Référence des AWS CLI commandes.

`describe-custom-routing-accelerator`

L'exemple de code suivant montre comment utiliser`describe-custom-routing-accelerator`.

AWS CLI

Pour décrire un accélérateur de routage personnalisé

L'`describe-custom-routing-accelerator`exemple suivant récupère les détails relatifs à l'accélérateur de routage personnalisé spécifié.

```
aws globalaccelerator describe-custom-routing-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleCustomRoutingAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
```

```
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847,
        "LastModifiedTime": 1542395013
    }
}
```

Pour plus d'informations, consultez la section Accélérateurs de routage personnalisés dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeCustomRoutingAcceleratorla section Référence des AWS CLI commandes.

`describe-custom-routing-endpoint-group`

L'exemple de code suivant montre comment utiliser`describe-custom-routing-endpoint-group`.

AWS CLI

Pour décrire un groupe de points de terminaison pour un accélérateur de routage personnalisé

L'`describe-custom-routing-endpoint-group`exemple suivant décrit un groupe de points de terminaison pour un accélérateur de routage personnalisé.

```
aws globalaccelerator describe-custom-routing-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab88888example
```

Sortie :

```
{
    "EndpointGroup": {
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab88888example",
        "EndpointGroupRegion": "us-east-2",
        "DestinationDescriptions": [
            {
                "FromPort": 5000,
                "ToPort": 10000,
```

```
                "Protocols": [
                    "UDP"
                ]
            }
        ],
        "EndpointDescriptions": [
            {
                "EndpointId": "subnet-1234567890abcdef0"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Groupes de points de terminaison pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeCustomRoutingEndpointGroupla section Référence des AWS CLI commandes.

`describe-custom-routing-listener`

L'exemple de code suivant montre comment utiliser`describe-custom-routing-listener`.

AWS CLI

Pour décrire un écouteur pour un accélérateur de routage personnalisé

L'`describe-custom-routing-listener`exemple suivant décrit un écouteur pour un accélérateur de routage personnalisé.

```
aws globalaccelerator describe-custom-routing-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Sortie :

```
{
    "Listener": {
        "PortRanges": [
            "FromPort": 5000,
            "ToPort": 10000
```

```
        ],
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234"
    }
}
```

Pour plus d'informations, consultez la section Écouteurs pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeCustomRoutingListenerla section Référence des AWS CLI commandes.

`describe-endpoint-group`

L'exemple de code suivant montre comment utiliser`describe-endpoint-group`.

AWS CLI

Pour décrire un groupe de points de terminaison

L'`describe-endpoint-group`exemple suivant extrait les détails d'un groupe de points de terminaison avec les points de terminaison suivants : une EC2 instance Amazon, un ALB et un NLB.

```
aws globalaccelerator describe-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab88888example
```

Sortie :

```
{
    "EndpointGroup": {
        "TrafficDialPercentage": 100.0,
        "EndpointDescriptions": [
        {
            "Weight": 128,
            "EndpointId": "i-1234567890abcdef0"
        },
```

```
        {
            "Weight": 128,
            "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
        },
        {
            "Weight": 128,
            "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
        }
        ],
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
        "EndpointGroupRegion": "us-east-1"
    }
}
```

Pour plus d'informations, consultez la section Groupes de terminaux dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeEndpointGroupla section Référence des AWS CLI commandes.

`describe-listener`

L'exemple de code suivant montre comment utiliser`describe-listener`.

AWS CLI

Pour décrire un auditeur

L'`describe-listener`exemple suivant décrit un écouteur.

```
aws globalaccelerator describe-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Sortie :

```
{
    "Listener": {
```

```
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
        "PortRanges": [
            {
                "FromPort": 80,
                "ToPort": 80
            }
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
}
```

Pour plus d'informations, voir Listeners in AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir DescribeListenerla section Référence des AWS CLI commandes.

`list-accelerators`

L'exemple de code suivant montre comment utiliser`list-accelerators`.

AWS CLI

Pour répertorier vos accélérateurs

L'`list-accelerators`exemple suivant répertorie les accélérateurs de votre AWS compte. Ce compte possède deux accélérateurs.

```
aws globalaccelerator list-accelerators
```

Sortie :

```
{
    "Accelerators": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
            "Name": "TestAccelerator",
            "IpAddressType": "IPV4",
```

```
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.250",
                        "198.51.100.52"
                    ]
                }
            ],
            "DnsName": "5a5a5a5a5a5a5a5a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1552424416.0,
            "LastModifiedTime": 1569375641.0
        },
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::888888888888:accelerator/8888abcd-abcd-8888-
abcd-8888EXAMPLE2",
            "Name": "ExampleAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.100",
                        "198.51.100.10"
                    ]
                }
            ],
            "DnsName": "6a6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

Pour plus d'informations, voir Accélérateurs dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListAcceleratorsla section Référence des AWS CLI commandes.

`list-byoip-cidr`

L'exemple de code suivant montre comment utiliser`list-byoip-cidr`.

AWS CLI

Pour répertorier vos plages d'adresses

L'`list-byoip-cidr`exemple suivant répertorie les plages d'adresses BYOIP (Bring your own IP address) que vous avez configurées pour être utilisées avec Global Accelerator.

```
aws globalaccelerator list-byoip-cidrs
```

Sortie :

```
{
    "ByoipCidrs": [
        {
            "Cidr": "198.51.100.0/24",
            "State": "READY"
        }
        {
            "Cidr": "203.0.113.25/24",
            "State": "READY"
        }
    ]
}
```

Pour plus d'informations, consultez la section Apportez votre propre adresse IP dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListByoipCidrla section Référence des AWS CLI commandes.

`list-custom-routing-accelerators`

L'exemple de code suivant montre comment utiliser`list-custom-routing-accelerators`.

AWS CLI

Pour répertorier vos accélérateurs de routage personnalisés

L'`list-custom-routing-accelerators`exemple suivant répertorie les accélérateurs de routage personnalisés d'un AWS compte.

```
aws globalaccelerator list-custom-routing-accelerators
```

Sortie :

```
{
    "Accelerators": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
            "Name": "TestCustomRoutingAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.250",
                        "198.51.100.52"
                    ]
                }
            ],
            "DnsName": "5a5a5a5a5a5a5a5a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1552424416.0,
            "LastModifiedTime": 1569375641.0
        },
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::888888888888:accelerator/8888abcd-abcd-8888-
abcd-8888EXAMPLE2",
            "Name": "ExampleCustomRoutingAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.100",
                        "198.51.100.10"
                    ]
                }
            ],
```

```
            "DnsName": "6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

Pour plus d'informations, consultez la section Accélérateurs de routage personnalisés dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListCustomRoutingAcceleratorsla section Référence des AWS CLI commandes.

`list-custom-routing-endpoint-groups`

L'exemple de code suivant montre comment utiliser`list-custom-routing-endpoint-groups`.

AWS CLI

Pour répertorier les groupes de points de terminaison d'un écouteur dans un accélérateur de routage personnalisé

L'`list-custom-routing-endpoint-groups`exemple suivant répertorie les groupes de points de terminaison d'un écouteur dans un accélérateur de routage personnalisé.

```
aws globalaccelerator list-custom-routing-endpoint-groups \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Sortie :

```
{
    "EndpointGroups": [
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
            "EndpointGroupRegion": "eu-central-1",
            "DestinationDescriptions": [
                {
                    "FromPort": 80,
```

```
                    "ToPort": 80,
                    "Protocols": [
                        "TCP",
                        "UDP"
                    ]
                }
            ]
            "EndpointDescriptions": [
                {
                    "EndpointId": "subnet-abcd123example"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes de points de terminaison pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListCustomRoutingEndpointGroupsla section Référence des AWS CLI commandes.

`list-custom-routing-listeners`

L'exemple de code suivant montre comment utiliser`list-custom-routing-listeners`.

AWS CLI

Pour répertorier les écouteurs des accélérateurs de routage personnalisés

L'`list-custom-routing-listeners`exemple suivant répertorie les écouteurs d'un accélérateur de routage personnalisé.

```
aws globalaccelerator list-custom-routing-listeners \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "Listeners": [
```

```
        {
            "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
            "PortRanges": [
                {
                    "FromPort": 5000,
                    "ToPort": 10000
                }
            ],
            "Protocol": "TCP"
        }
    ]
}
```

Pour plus d'informations, consultez la section Écouteurs pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListCustomRoutingListenersla section Référence des AWS CLI commandes.

`list-custom-routing-port-mappings-by-destination`

L'exemple de code suivant montre comment utiliser`list-custom-routing-port-mappings-by-destination`.

AWS CLI

Pour répertorier les mappages de ports pour une destination d'accélérateur de routage personnalisée spécifique

L'`list-custom-routing-port-mappings-by-destination`exemple suivant fournit les mappages de ports pour un EC2 serveur de destination spécifique (à l'adresse de destination) pour un accélérateur de routage personnalisé.

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \
    --endpoint-id subnet-abcd123example \
    --destination-address 198.51.100.52
```

Sortie :

```
{
    "DestinationPortMappings": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::402092451327:accelerator/24ea29b8-
d750-4489-8919-3095f3c4b0a7",
                "AcceleratorSocketAddresses": [
                    {
                        "IpAddress": "192.0.2.250",
                        "Port": 65514
                    },
                    {
                        "IpAddress": "192.10.100.99",
                        "Port": 65514
                    }
                ],
                "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example",
                "EndpointId": "subnet-abcd123example",
                "EndpointGroupRegion": "us-west-2",
                "DestinationSocketAddress": {
                    "IpAddress": "198.51.100.52",
                    "Port": 80
                },
                "IpAddressType": "IPv4",
                "DestinationTrafficState": "ALLOW"
        }
    ]
}
```

Pour plus d'informations, consultez la section Fonctionnement des accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListCustomRoutingPortMappingsByDestinationla section Référence des AWS CLI commandes.

`list-custom-routing-port-mappings`

L'exemple de code suivant montre comment utiliser`list-custom-routing-port-mappings`.

AWS CLI

Pour répertorier les mappages de ports dans un accélérateur de routage personnalisé

L'`list-custom-routing-port-mappings`exemple suivant fournit une liste partielle des mappages de ports dans un accélérateur de routage personnalisé.

```
aws globalaccelerator list-custom-routing-port-mappings \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "PortMappings": [
        {
            "AcceleratorPort": 40480,
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
            "EndpointId": "subnet-1234567890abcdef0",
            "DestinationSocketAddress": {
                "IpAddress": "192.0.2.250",
                "Port": 80
            },
            "Protocols": [
                "TCP",
                "UDP"
            ],
            "DestinationTrafficState": "ALLOW"
        }
        {
            "AcceleratorPort": 40481,
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
            "EndpointId": "subnet-1234567890abcdef0",
            "DestinationSocketAddress": {
               "IpAddress": "192.0.2.251",
              "Port": 80
            },
            "Protocols": [
                "TCP",
```

```
                "UDP"
            ],
            "DestinationTrafficState": "ALLOW"
        }
    ]
}
```

Pour plus d'informations, consultez la section Fonctionnement des accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListCustomRoutingPortMappingsla section Référence des AWS CLI commandes.

`list-endpoint-groups`

L'exemple de code suivant montre comment utiliser`list-endpoint-groups`.

AWS CLI

Pour répertorier les groupes de terminaux

L'`list-endpoint-groups`exemple suivant répertorie les groupes de points de terminaison d'un écouteur. Cet écouteur possède deux groupes de points de terminaison.

```
aws globalaccelerator --region us-west-2 list-endpoint-groups \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Sortie :

```
{
    "EndpointGroups": [
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
            "EndpointGroupRegion": "eu-central-1",
            "EndpointDescriptions": [],
            "TrafficDialPercentage": 100.0,
            "HealthCheckPort": 80,
```

```
            "HealthCheckProtocol": "TCP",
            "HealthCheckIntervalSeconds": 30,
            "ThresholdCount": 3
        }
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab99999example",
            "EndpointGroupRegion": "us-east-1",
            "EndpointDescriptions": [],
            "TrafficDialPercentage": 50.0,
            "HealthCheckPort": 80,
            "HealthCheckProtocol": "TCP",
            "HealthCheckIntervalSeconds": 30,
            "ThresholdCount": 3
        }
    ]
}
```

Pour plus d'informations, consultez la section Endpoint Groups in AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListEndpointGroupsla section Référence des AWS CLI commandes.

`list-listeners`

L'exemple de code suivant montre comment utiliser`list-listeners`.

AWS CLI

Pour répertorier les auditeurs

L'`list-listeners`exemple suivant répertorie les écouteurs d'un accélérateur.

```
aws globalaccelerator list-listeners \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
```

```
    "Listeners": [
        {
            "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
            "PortRanges": [
                {
                    "FromPort": 80,
                    "ToPort": 80
                }
            ],
            "Protocol": "TCP",
            "ClientAffinity": "NONE"
        }
    ]
}
```

Pour plus d'informations, voir Listeners in AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListListenersla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un accélérateur

L'`list-tags-for-resource`exemple suivant répertorie les balises d'un accélérateur spécifique.

```
aws globalaccelerator list-tags-for-resource \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Sortie :

```
{
    "Tags": [
        {
```

```
            "Key": "Project",
            "Value": "A123456"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`provision-byoip-cidr`

L'exemple de code suivant montre comment utiliser`provision-byoip-cidr`.

AWS CLI

Pour provisionner une plage d'adresses

L'`provision-byoip-cidr`exemple suivant indique la plage d'adresses spécifiée à utiliser avec vos AWS ressources.

```
aws globalaccelerator provision-byoip-cidr \
    --cidr 192.0.2.250/24 \
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "192.0.2.250/24",
        "State": "PENDING_PROVISIONING"
    }
}
```

Pour plus d'informations, consultez la section Apportez votre propre adresse IP dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir ProvisionByoipCidrla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un accélérateur

L'`tag-resource`exemple suivant ajoute les balises Name et Project à un accélérateur, ainsi que les valeurs correspondantes pour chacune d'elles.

```
aws globalaccelerator tag-resource \
    --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une étiquette d'un accélérateur

L'`untag-resource`exemple suivant supprime les balises Name et Project d'un accélérateur.

```
aws globalaccelerator untag-resource \
    --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --tag-keys Key="Name" Key="Project"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-accelerator-attributes`

L'exemple de code suivant montre comment utiliser`update-accelerator-attributes`.

AWS CLI

Pour mettre à jour les attributs d'un accélérateur

L'`update-accelerator-attributes`exemple suivant met à jour un accélérateur pour activer les journaux de flux. Vous devez spécifier la `US-West-2 (Oregon)` région pour créer ou mettre à jour les attributs de l'accélérateur.

```
aws globalaccelerator update-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --flow-logs-enabled \
    --flow-logs-s3-bucket flowlogs-abc \
    --flow-logs-s3-prefix bucketprefix-abc
```

Sortie :

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

Pour plus d'informations, voir Accélérateurs dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateAcceleratorAttributesla section Référence des AWS CLI commandes.

`update-accelerator`

L'exemple de code suivant montre comment utiliser`update-accelerator`.

AWS CLI

Pour mettre à jour un accélérateur

L'`update-accelerator`exemple suivant modifie un accélérateur pour en changer le nom en`ExampleAcceleratorNew`. Vous devez spécifier la `US-West-2 (Oregon)` région pour créer ou mettre à jour les accélérateurs.

```
aws globalaccelerator update-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --name ExampleAcceleratorNew
```

Sortie :

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleAcceleratorNew",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1232394847,
        "LastModifiedTime": 1232395654
    }
}
```

Pour plus d'informations, voir Accélérateurs dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateAcceleratorla section Référence des AWS CLI commandes.

`update-custom-routing-accelerator-attributes`

L'exemple de code suivant montre comment utiliser`update-custom-routing-accelerator-attributes`.

AWS CLI

Pour mettre à jour les attributs d'un accélérateur de routage personnalisé

L'`update-custom-routing-accelerator-attributes`exemple suivant met à jour un accélérateur de routage personnalisé pour activer les journaux de flux.

```
aws globalaccelerator update-custom-routing-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --flow-logs-enabled \
    --flow-logs-s3-bucket flowlogs-abc \
    --flow-logs-s3-prefix bucketprefix-abc
```

Sortie :

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

Pour plus d'informations, consultez la section Accélérateurs de routage personnalisés dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateCustomRoutingAcceleratorAttributesla section Référence des AWS CLI commandes.

`update-custom-routing-accelerator`

L'exemple de code suivant montre comment utiliser`update-custom-routing-accelerator`.

AWS CLI

Pour mettre à jour un accélérateur de routage personnalisé

L'`update-custom-routing-accelerator`exemple suivant modifie un accélérateur de routage personnalisé pour changer le nom de l'accélérateur.

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --name ExampleCustomRoutingAcceleratorNew
```

Sortie :

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleCustomRoutingAcceleratorNew",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1232394847,
        "LastModifiedTime": 1232395654
    }
}
```

Pour plus d'informations, consultez la section Accélérateurs de routage personnalisés dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateCustomRoutingAcceleratorla section Référence des AWS CLI commandes.

`update-custom-routing-listener`

L'exemple de code suivant montre comment utiliser`update-custom-routing-listener`.

AWS CLI

Pour mettre à jour un écouteur pour un accélérateur de routage personnalisé

L'`update-custom-routing-listener`exemple suivant met à jour un écouteur pour modifier la plage de ports.

```
aws globalaccelerator update-custom-routing-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --port-ranges FromPort=10000,ToPort=20000
```

Sortie :

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz
        "PortRanges": [
            {
                "FromPort": 10000,
                "ToPort": 20000
            }
        ],
        "Protocol": "TCP"
    }
}
```

Pour plus d'informations, consultez la section Écouteurs pour les accélérateurs de routage personnalisés dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateCustomRoutingListenerla section Référence des AWS CLI commandes.

`update-endpoint-group`

L'exemple de code suivant montre comment utiliser`update-endpoint-group`.

AWS CLI

Pour mettre à jour un groupe de terminaux

L'`update-endpoint-group`exemple suivant ajoute trois points de terminaison à un groupe de points de terminaison : une adresse IP élastique, un ALB et un NLB.

```
aws globalaccelerator update-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab88888example \
    --endpoint-configurations \
        EndpointId=eipalloc-eip01234567890abc,Weight=128 \
        EndpointId=arn:aws:elasticloadbalancing:us-east-1:000123456789:loadbalancer/
app/ALBTesting/alb01234567890xyz,Weight=128 \
        EndpointId=arn:aws:elasticloadbalancing:us-east-1:000123456789:loadbalancer/
net/NLBTesting/alb01234567890qrs,Weight=128
```

Sortie :

```
{
    "EndpointGroup": {
        "TrafficDialPercentage": 100,
        "EndpointDescriptions": [
            {
                "Weight": 128,
                "EndpointId": "eip01234567890abc"
            },
            {
                "Weight": 128,
                "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
            },
            {
                "Weight": 128,
                "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
            }
```

```
        ],
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
        "EndpointGroupRegion": "us-east-1"
    }
}
```

Pour plus d'informations, consultez la section Groupes de terminaux dans AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateEndpointGroupla section Référence des AWS CLI commandes.

`update-listener`

L'exemple de code suivant montre comment utiliser`update-listener`.

AWS CLI

Pour mettre à jour un écouteur

L'`update-listener`exemple suivant met à jour un écouteur pour changer le port à 100.

```
aws globalaccelerator update-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --port-ranges FromPort=100,ToPort=100
```

Sortie :

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz
        "PortRanges": [
            {
                "FromPort": 100,
                "ToPort": 100
            }
```

```
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
}
```

Pour plus d'informations, voir Listeners in AWS Global Accelerator dans le Guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir UpdateListenerla section Référence des AWS CLI commandes.

`withdraw-byoip-cidr`

L'exemple de code suivant montre comment utiliser`withdraw-byoip-cidr`.

AWS CLI

Pour retirer une plage d'adresses

L'`withdraw-byoip-cidr`exemple suivant retire de AWS Global Accelerator une plage d'adresses que vous avez précédemment annoncée pour une utilisation avec vos AWS ressources.

```
aws globalaccelerator withdraw-byoip-cidr \
    --cidr 192.0.2.250/24
```

Sortie :

```
{
    "ByoipCidr": {
        "Cidr": "192.0.2.250/24",
        "State": "PENDING_WITHDRAWING"
    }
}
```

Pour plus d'informations, consultez la section Apportez votre propre adresse IP dans AWS Global Accelerator dans le guide du développeur de AWS Global Accelerator.

- Pour plus de détails sur l'API, voir WithdrawByoipCidrla section Référence des AWS CLI commandes.

AWS Glue exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Glue.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-stop-job-run`

L'exemple de code suivant montre comment utiliser`batch-stop-job-run`.

AWS CLI

Pour arrêter les exécutions de tâches

L'`batch-stop-job-run`exemple suivant arrête l'exécution d'une tâche.

```
aws glue batch-stop-job-run \
    --job-name "my-testing-job" \
    --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```

Sortie :

```
{
    "SuccessfulSubmissions": [
        {
            "JobName": "my-testing-job",
            "JobRunId":
 "jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f"
        }
    ],
```

```
    "Errors": [],
    "ResponseMetadata": {
        "RequestId": "66bd6b90-01db-44ab-95b9-6aeff0e73d88",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
            "date": "Fri, 16 Oct 2020 20:54:51 GMT",
            "content-type": "application/x-amz-json-1.1",
            "content-length": "148",
            "connection": "keep-alive",
            "x-amzn-requestid": "66bd6b90-01db-44ab-95b9-6aeff0e73d88"
        },
        "RetryAttempts": 0
    }
}
```

Pour plus d'informations, consultez Exécutions de tâches dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous BatchStopJobRunà la section Référence des AWS CLI commandes.

`create-connection`

L'exemple de code suivant montre comment utiliser`create-connection`.

AWS CLI

Pour créer une connexion pour les magasins AWS de données Glue

L'`create-connection`exemple suivant crée une connexion dans le catalogue de données AWS Glue qui fournit des informations de connexion pour un magasin de données Kafka.

```
aws glue create-connection \
    --connection-input '{ \
        "Name":"conn-kafka-custom", \
        "Description":"kafka connection with ssl to custom kafka", \
        "ConnectionType":"KAFKA",  \
        "ConnectionProperties":{  \
            "KAFKA_BOOTSTRAP_SERVERS":"<Kafka-broker-server-url>:<SSL-Port>", \
            "KAFKA_SSL_ENABLED":"true", \
            "KAFKA_CUSTOM_CERT": "s3://bucket/prefix/cert-file.pem" \
        }, \
        "PhysicalConnectionRequirements":{ \
```

```
            "SubnetId":"subnet-1234", \
            "SecurityGroupIdList":["sg-1234"], \
            "AvailabilityZone":"us-east-1a"} \
    }' \
    --region us-east-1
    --endpoint https://glue.us-east-1.amazonaws.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Defining Connections in the AWS Glue Data Catalog du AWS Glue Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateConnectionà la section Référence des AWS CLI commandes.

`create-database`

L'exemple de code suivant montre comment utiliser`create-database`.

AWS CLI

Pour créer une base de données

L'`create-database`exemple suivant crée une base de données dans le catalogue de données AWS Glue.

```
aws glue create-database \
    --database-input "{\"Name\":\"tempdb\"}" \
    --profile my_profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Définition d'une base de données dans votre catalogue de données dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous CreateDatabaseà la section Référence des AWS CLI commandes.

`create-job`

L'exemple de code suivant montre comment utiliser`create-job`.

AWS CLI

Pour créer une tâche afin de transformer des données

L'exemple `create-job` suivant crée une tâche de streaming qui exécute un script stocké dans S3.

```
aws glue create-job \
    --name my-testing-job \
    --role AWSGlueServiceRoleDefault \
    --command '{ \
        "Name": "gluestreaming", \
        "ScriptLocation": "s3://amzn-s3-demo-bucket/folder/" \
    }' \
    --region us-east-1 \
    --output json \
    --default-arguments '{ \
        "--job-language":"scala", \
        "--class":"GlueApp" \
    }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

Contenu de `test_script.scala` :

```
import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
    def main(sysArgs: Array[String]) {
        val spark: SparkContext = new SparkContext()
        val glueContext: GlueContext = new GlueContext(spark)
        // @params: [JOB_NAME]
        val args = GlueArgParser.getResolvedOptions(sysArgs,
 Seq("JOB_NAME").toArray)
```

```
        Job.init(args("JOB_NAME"), glueContext, args.asJava)
        // @type: DataSource
        // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx
 = "datasource0"]
        // @return: datasource0
        // @inputs: []
        val datasource0 = glueContext.getCatalogSource(database = "tempdb",
 tableName = "s3-source", redshiftTmpDir = "", transformationContext =
 "datasource0").getDynamicFrame()
        // @type: ApplyMapping
        // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
 ("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
 "status", "string")], transformation_ctx = "applymapping1"]
        // @return: applymapping1
        // @inputs: [frame = datasource0]
        val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
 "int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
 "int"), ("status", "string", "status", "string")), caseSensitive = false,
 transformationContext = "applymapping1")
        // @type: SelectFields
        // @args: [paths = ["sensorid", "currenttemperature", "status"],
 transformation_ctx = "selectfields2"]
        // @return: selectfields2
        // @inputs: [frame = applymapping1]
        val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
 "currenttemperature", "status"), transformationContext = "selectfields2")
        // @type: ResolveChoice
        // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-
s3-sink", transformation_ctx = "resolvechoice3"]
        // @return: resolvechoice3
        // @inputs: [frame = selectfields2]
        val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
 Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
 Some("my-s3-sink"), transformationContext = "resolvechoice3")
        // @type: DataSink
        // @args: [database = "tempdb", table_name = "my-s3-sink",
 transformation_ctx = "datasink4"]
        // @return: datasink4
        // @inputs: [frame = resolvechoice3]
        val datasink4 = glueContext.getCatalogSink(database = "tempdb",
 tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
 "datasink4").writeDynamicFrame(resolvechoice3)
        Job.commit()
    }
```

```
}
```

Sortie :

```
{
    "Name": "my-testing-job"
}
```

Pour plus d'informations, consultez la section Création de jobs dans AWS Glue du AWS Glue Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateJobà la section Référence des AWS CLI commandes.

`create-table`

L'exemple de code suivant montre comment utiliser`create-table`.

AWS CLI

Exemple 1 : pour créer une table pour un flux de données Kinesis

L'`create-table`exemple suivant crée une table dans le catalogue de données AWS Glue qui décrit un flux de données Kinesis.

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"test-kinesis-input", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"sensorid", "Type":"int"}, \
                {"Name":"currenttemperature", "Type":"int"}, \
                {"Name":"status", "Type":"string"} \
            ], \
            "Location":"my-testing-stream", \
            "Parameters":{ \
                "typeOfData":"kinesis","streamName":"my-testing-stream", \
                "kinesisUrl":"https://kinesis.us-east-1.amazonaws.com" \
            }, \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.openx.data.jsonserde.JsonSerDe"} \
        }, \
        "Parameters":{ \
            "classification":"json"} \
```

```
        }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition des tables dans le catalogue de données AWS Glue du AWS Glue Developer Guide.

Exemple 2 : pour créer une table pour un magasin de données Kafka

L'`create-table`exemple suivant crée une table dans le catalogue de données AWS Glue qui décrit un magasin de données Kafka.

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"test-kafka-input", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"sensorid", "Type":"int"}, \
                {"Name":"currenttemperature", "Type":"int"}, \
                {"Name":"status", "Type":"string"} \
            ], \
            "Location":"glue-topic", \
            "Parameters":{ \
                "typeOfData":"kafka","topicName":"glue-topic", \
                "connectionName":"my-kafka-connection" \
            }, \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.apache.hadoop.hive.serde2.OpenCSVSerde"}
 \
        }, \
        "Parameters":{ \
            "separatorChar":","} \
        }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition des tables dans le catalogue de données AWS Glue du AWS Glue Developer Guide.

Exemple 3 : pour créer une table pour un magasin de données AWS S3

L'`create-table`exemple suivant crée une table dans le catalogue de données AWS Glue qui décrit un magasin de données AWS Simple Storage Service (AWS S3).

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"s3-output", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"s1", "Type":"string"}, \
                {"Name":"s2", "Type":"int"}, \
                {"Name":"s3", "Type":"string"} 
            ], \
            "Location":"s3://bucket-path/", \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.openx.data.jsonserde.JsonSerDe"} \
        }, \
        "Parameters":{ \
            "classification":"json"} \
        }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition des tables dans le catalogue de données AWS Glue du AWS Glue Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateTableà la section Référence des AWS CLI commandes.

`delete-job`

L'exemple de code suivant montre comment utiliser`delete-job`.

AWS CLI

Pour supprimer une tâche

L'exemple `delete-job` suivant supprime une tâche qui n'est plus nécessaire.

```
aws glue delete-job \
    --job-name my-testing-job
```

Sortie :

```
{
    "JobName": "my-testing-job"
}
```

Pour plus d'informations, consultez Working with Jobs on the AWS Glue Console dans le AWS Glue Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteJobà la section Référence des AWS CLI commandes.

`get-databases`

L'exemple de code suivant montre comment utiliser`get-databases`.

AWS CLI

Pour répertorier les définitions de certaines ou de toutes les bases de données du AWS Glue Data Catalog

L'exemple `get-databases` suivant renvoie des informations sur les bases de données du catalogue de données.

```
aws glue get-databases
```

Sortie :

```
{
    "DatabaseList": [
        {
            "Name": "default",
            "Description": "Default Hive database",
            "LocationUri": "file:/spark-warehouse",
            "CreateTime": 1602084052.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
```

```
                ]
            }
        ],
        "CatalogId": "111122223333"
    },
    {
        "Name": "flights-db",
        "CreateTime": 1587072847.0,
        "CreateTableDefaultPermissions": [
            {
                "Principal": {
                    "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                },
                "Permissions": [
                    "ALL"
                ]
            }
        ],
        "CatalogId": "111122223333"
    },
    {
        "Name": "legislators",
        "CreateTime": 1601415625.0,
        "CreateTableDefaultPermissions": [
            {
                "Principal": {
                    "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                },
                "Permissions": [
                    "ALL"
                ]
            }
        ],
        "CatalogId": "111122223333"
    },
    {
        "Name": "tempdb",
        "CreateTime": 1601498566.0,
        "CreateTableDefaultPermissions": [
            {
                "Principal": {
                    "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                },
                "Permissions": [
```

```
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        }
    ]
}
```

Pour plus d'informations, consultez Définition d'une base de données dans votre catalogue de données dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous GetDatabasesà la section Référence des AWS CLI commandes.

`get-job-run`

L'exemple de code suivant montre comment utiliser`get-job-run`.

AWS CLI

Pour obtenir des informations sur l'exécution d'une tâche

L'exemple `get-job-run` suivant récupère des informations sur l'exécution d'une tâche.

```
aws glue get-job-run \
    --job-name "Combine legistators data" \
    --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

Sortie :

```
{
    "JobRun": {
        "Id": "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
        "Attempt": 0,
        "JobName": "Combine legistators data",
        "StartedOn": 1602873931.255,
        "LastModifiedOn": 1602874075.985,
        "CompletedOn": 1602874075.985,
        "JobRunState": "SUCCEEDED",
        "Arguments": {
            "--enable-continuous-cloudwatch-log": "true",
            "--enable-metrics": "",
```

```
            "--enable-spark-ui": "true",
            "--job-bookmark-option": "job-bookmark-enable",
            "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-east-1/
sparkHistoryLogs/"
        },
        "PredecessorRuns": [],
        "AllocatedCapacity": 10,
        "ExecutionTime": 117,
        "Timeout": 2880,
        "MaxCapacity": 10.0,
        "WorkerType": "G.1X",
        "NumberOfWorkers": 10,
        "LogGroupName": "/aws-glue/jobs",
        "GlueVersion": "2.0"
    }
}
```

Pour plus d'informations, consultez Exécutions de tâches dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous GetJobRunà la section Référence des AWS CLI commandes.

`get-job-runs`

L'exemple de code suivant montre comment utiliser`get-job-runs`.

AWS CLI

Pour obtenir des informations sur toutes les exécutions d'une tâche

L'exemple `get-job-runs` suivant récupère des informations sur toutes les exécutions d'une tâche.

```
aws glue get-job-runs \
    --job-name "my-testing-job"
```

Sortie :

```
{
    "JobRuns": [
        {
```

```
            "Id":
 "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
            "Attempt": 0,
            "JobName": "my-testing-job",
            "StartedOn": 1602873931.255,
            "LastModifiedOn": 1602874075.985,
            "CompletedOn": 1602874075.985,
            "JobRunState": "SUCCEEDED",
            "Arguments": {
                "--enable-continuous-cloudwatch-log": "true",
                "--enable-metrics": "",
                "--enable-spark-ui": "true",
                "--job-bookmark-option": "job-bookmark-enable",
                "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-
east-1/sparkHistoryLogs/"
            },
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 117,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        },
        {
            "Id":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_2",
            "Attempt": 2,
            "PreviousRunId":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
            "JobName": "my-testing-job",
            "StartedOn": 1602811168.496,
            "LastModifiedOn": 1602811282.39,
            "CompletedOn": 1602811282.39,
            "JobRunState": "FAILED",
            "ErrorMessage": "An error occurred while calling
 o122.pyWriteDynamicFrame.
                Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied;
                Request ID: 021AAB703DB20A2D;
                S3 Extended Request ID: teZk24Y09TkXzBvMPG502L5VJBhe9DJuWA9/
TXtuGOqfByajkfL/Tlqt5JBGdEGpigAqzdMDM/U=)",
```

```
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 110,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        },
        {
            "Id":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
            "Attempt": 1,
            "PreviousRunId":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f",
            "JobName": "my-testing-job",
            "StartedOn": 1602811020.518,
            "LastModifiedOn": 1602811138.364,
            "CompletedOn": 1602811138.364,
            "JobRunState": "FAILED",
            "ErrorMessage": "An error occurred while calling
 o122.pyWriteDynamicFrame.
                Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied;
                Request ID: 2671D37856AE7ABB;
                S3 Extended Request ID: RLJCJw20brV
+PpC6GpORahyF2fp9flB5SSb2bTGPnUSPVizLXRl1PN3QZldb+v1o9qRVktNYbW8=)",
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 113,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        }
    ]
}
```

Pour plus d'informations, consultez Exécutions de tâches dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous GetJobRunsà la section Référence des AWS CLI commandes.

`get-job`

L'exemple de code suivant montre comment utiliser`get-job`.

AWS CLI

Pour récupérer des informations sur une tâche

L'exemple `get-job` suivant récupère des informations sur une tâche.

```
aws glue get-job \
    --job-name my-testing-job
```

Sortie :

```
{
    "Job": {
        "Name": "my-testing-job",
        "Role": "Glue_DefaultRole",
        "CreatedOn": 1602805698.167,
        "LastModifiedOn": 1602805698.167,
        "ExecutionProperty": {
            "MaxConcurrentRuns": 1
        },
        "Command": {
            "Name": "gluestreaming",
            "ScriptLocation": "s3://janetst-bucket-01/Scripts/test_script.scala",
            "PythonVersion": "2"
        },
        "DefaultArguments": {
            "--class": "GlueApp",
            "--job-language": "scala"
        },
        "MaxRetries": 0,
        "AllocatedCapacity": 10,
        "MaxCapacity": 10.0,
        "GlueVersion": "1.0"
    }
}
```

Pour plus d'informations, consultez Tâches dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous GetJobà la section Référence des AWS CLI commandes.

`get-plan`

L'exemple de code suivant montre comment utiliser`get-plan`.

AWS CLI

Pour obtenir le code généré pour mapper les données des tables sources vers les tables cibles

Ce qui suit `get-plan` extrait le code généré pour mapper les colonnes de la source de données à la cible de données.

```
aws glue get-plan --mapping '[ \
    { \
        "SourcePath":"sensorid", \
        "SourceTable":"anything", \
        "SourceType":"int", \
        "TargetPath":"sensorid", \
        "TargetTable":"anything", \
        "TargetType":"int" \
    }, \
    { \
        "SourcePath":"currenttemperature", \
        "SourceTable":"anything", \
        "SourceType":"int", \
        "TargetPath":"currenttemperature", \
        "TargetTable":"anything", \
        "TargetType":"int" \
    }, \
    { \
        "SourcePath":"status", \
        "SourceTable":"anything", \
        "SourceType":"string", \
        "TargetPath":"status", \
        "TargetTable":"anything", \
        "TargetType":"string" \
    }]' \
    --source '{ \
        "DatabaseName":"tempdb", \
```

```
        "TableName":"s3-source" \
    }' \
    --sinks '[ \
        { \
            "DatabaseName":"tempdb", \
            "TableName":"my-s3-sink" \
        }]' 
    --language "scala"
    --endpoint https://glue.us-east-1.amazonaws.com
    --output "text"
```

Sortie :

```
import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // @type: DataSource
    // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx =
 "datasource0"]
    // @return: datasource0
    // @inputs: []
    val datasource0 = glueContext.getCatalogSource(database = "tempdb",
 tableName = "s3-source", redshiftTmpDir = "", transformationContext =
 "datasource0").getDynamicFrame()
    // @type: ApplyMapping
    // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
 ("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
 "status", "string")], transformation_ctx = "applymapping1"]
```

```
    // @return: applymapping1
    // @inputs: [frame = datasource0]
    val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
 "int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
 "int"), ("status", "string", "status", "string")), caseSensitive = false,
 transformationContext = "applymapping1")
    // @type: SelectFields
    // @args: [paths = ["sensorid", "currenttemperature", "status"],
 transformation_ctx = "selectfields2"]
    // @return: selectfields2
    // @inputs: [frame = applymapping1]
    val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
 "currenttemperature", "status"), transformationContext = "selectfields2")
    // @type: ResolveChoice
    // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-s3-
sink", transformation_ctx = "resolvechoice3"]
    // @return: resolvechoice3
    // @inputs: [frame = selectfields2]
    val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
 Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
 Some("my-s3-sink"), transformationContext = "resolvechoice3")
    // @type: DataSink
    // @args: [database = "tempdb", table_name = "my-s3-sink", transformation_ctx =
 "datasink4"]
    // @return: datasink4
    // @inputs: [frame = resolvechoice3]
    val datasink4 = glueContext.getCatalogSink(database = "tempdb",
 tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
 "datasink4").writeDynamicFrame(resolvechoice3)
    Job.commit()
  }
}
```

Pour plus d'informations, consultez la section Modification de scripts dans AWS Glue du manuel AWS Glue Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetPlanà la section Référence des AWS CLI commandes.

`get-tables`

L'exemple de code suivant montre comment utiliser`get-tables`.

AWS CLI

Pour répertorier les définitions de tout ou partie des tables dans la base de données spécifiée

L'exemple `get-tables` suivant renvoie des informations sur les tables de la base de données spécifiée.

```
aws glue get-tables --database-name 'tempdb'
```

Sortie :

```
{
    "TableList": [
        {
            "Name": "my-s3-sink",
            "DatabaseName": "tempdb",
            "CreateTime": 1602730539.0,
            "UpdateTime": 1602730539.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "s3://janetst-bucket-01/test-s3-output/",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SerdeInfo": {
                    "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
                },
                "SortColumns": [],
                "StoredAsSubDirectories": false
            },
```

```
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        },
        {
            "Name": "s3-source",
            "DatabaseName": "tempdb",
            "CreateTime": 1602730658.0,
            "UpdateTime": 1602730658.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "s3://janetst-bucket-01/",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SortColumns": [],
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        },
        {
            "Name": "test-kinesis-input",
            "DatabaseName": "tempdb",
```

```
            "CreateTime": 1601507001.0,
            "UpdateTime": 1601507001.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "my-testing-stream",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SerdeInfo": {
                    "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
                },
                "SortColumns": [],
                "Parameters": {
                    "kinesisUrl": "https://kinesis.us-east-1.amazonaws.com",
                    "streamName": "my-testing-stream",
                    "typeOfData": "kinesis"
                },
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        }
    ]
}
```

Pour plus d'informations, consultez la section Définition des tables dans le catalogue de données AWS Glue du AWS Glue Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetTablesà la section Référence des AWS CLI commandes.

`start-crawler`

L'exemple de code suivant montre comment utiliser`start-crawler`.

AWS CLI

Pour démarrer un crawler

L'exemple `start-crawler` suivant démarre un crawler.

```
aws glue start-crawler --name my-crawler
```

Sortie :

```
None
```

Pour plus d'informations, consultez Définition des crawlers dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous StartCrawlerà la section Référence des AWS CLI commandes.

`start-job-run`

L'exemple de code suivant montre comment utiliser`start-job-run`.

AWS CLI

Pour commencer l'exécution d'une tâche

L'exemple `start-job-run` suivant démarre une tâche.

```
aws glue start-job-run \
    --job-name my-job
```

Sortie :

```
{
    "JobRunId":
 "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"
}
```

Pour plus d'informations, consultez Création de tâches dans le Guide du développeur AWS Glue.

- Pour plus de détails sur l'API, reportez-vous StartJobRunà la section Référence des AWS CLI commandes.

GuardDuty exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with GuardDuty.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-invitation`

L'exemple de code suivant montre comment utiliser`accept-invitation`.

AWS CLI

Pour accepter une invitation à devenir GuardDuty membre dans la région actuelle

L'`accept-invitation`exemple suivant montre comment accepter une invitation à devenir un compte GuardDuty membre dans la région actuelle.

```
aws guardduty accept-invitation  \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
```

```
    --master-id 123456789111 \
    --invitation-id d6b94fb03a66ff665f7db8764example
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion GuardDuty des comptes sur invitation dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous AcceptInvitationà la section Référence des AWS CLI commandes.

`archive-findings`

L'exemple de code suivant montre comment utiliser`archive-findings`.

AWS CLI

Pour archiver les résultats dans la région actuelle

Cet `archive-findings` exemple montre comment archiver les résultats dans la région actuelle.

```
aws guardduty archive-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création de règles de suppression dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ArchiveFindingsà la section Référence des AWS CLI commandes.

`create-detector`

L'exemple de code suivant montre comment utiliser`create-detector`.

AWS CLI

Pour activer GuardDuty dans la région actuelle

Cet exemple montre comment créer un nouveau détecteur, qui active GuardDuty, dans la région actuelle. :

```
aws guardduty create-detector \
    --enable
```

Sortie :

```
{
    "DetectorId": "b6b992d6d2f48e64bc59180bfexample"
}
```

Pour plus d'informations, consultez la section Activer Amazon GuardDuty dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateDetectorà la section Référence des AWS CLI commandes.

`create-filter`

L'exemple de code suivant montre comment utiliser`create-filter`.

AWS CLI

Exemple 1 : pour créer un nouveau filtre dans la région actuelle

L'`create-filter`exemple suivant crée un filtre qui correspond à tous les résultats de Portscan, créés par exemple à partir d'une image spécifique. Cela ne supprime pas ces résultats.

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myFilterExample \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

Sortie :

```
{
    "Name": "myFilterExample"
}
```

Pour plus d'informations, consultez la section GuardDuty Résultats du filtrage dans le Guide de GuardDuty l'utilisateur.

Exemple 2 : pour créer un nouveau filtre et supprimer les résultats dans la région actuelle

L'`create-filter`exemple suivant crée un filtre qui correspond à tous les résultats de Portscan, créés par exemple à partir d'une image spécifique. Ce filtre archive ces résultats afin qu'ils n'apparaissent pas dans vos résultats actuels.

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --action ARCHIVE \
    --name myFilterSecondExample \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

Sortie :

```
{
    "Name": "myFilterSecondExample"
}
```

Pour plus d'informations, consultez la section GuardDuty Résultats du filtrage dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateFilterà la section Référence des AWS CLI commandes.

`create-ip-set`

L'exemple de code suivant montre comment utiliser`create-ip-set`.

AWS CLI

Pour créer et activer un ensemble d'adresses IP fiables

L'`create-ip-set`exemple suivant crée et active un ensemble d'adresses IP fiables dans la région actuelle.

```
aws guardduty create-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --name new-ip-set-example \
    --format TXT \
    --location s3://amzn-s3-demo-bucket/customtrustlist.csv \
```

```
    --activate
```

Sortie :

```
{
    "IpSetId": "d4b94fc952d6912b8f3060768example"
}
```

Pour plus d'informations, consultez la section Utilisation de listes d'adresses IP fiables et de listes de menaces dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateIpSetà la section Référence des AWS CLI commandes.

`create-members`

L'exemple de code suivant montre comment utiliser`create-members`.

AWS CLI

Pour associer un nouveau membre à votre compte GuardDuty principal dans la région actuelle.

Cet exemple montre comment associer des comptes de membres à gérer par le compte courant en tant que compte GuardDuty principal.

```
aws guardduty create-members
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

Sortie :

```
{
   "UnprocessedAccounts": []
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateMembersà la section Référence des AWS CLI commandes.

`create-publishing-destination`

L'exemple de code suivant montre comment utiliser`create-publishing-destination`.

AWS CLI

Pour créer une destination de publication vers laquelle exporter GuardDuty les résultats de la région actuelle.

L'`create-publishing-destination`exemple suivant montre comment configurer une destination de publication pour exporter les GuardDuty résultats actuels (et non archivés) afin de suivre les données des résultats historiques.

```
aws guardduty create-publishing-destination \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --destination-type S3 \
    --destination-properties 'DestinationArn=arn:aws:s3:::amzn-s3-demo-bucket,KmsKeyArn=arn:aws:kms:us-west-1:111122223333:key/84cee9c5-dea1-401a-ab6d-e1de7example'
```

Sortie :

```
{
    "DestinationId": "46b99823849e1bbc242dfbe3cexample"
}
```

Pour plus d'informations, consultez la section Exportation des GuardDuty résultats générés vers des compartiments Amazon S3 dans le guide de l'GuardDuty utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreatePublishingDestinationà la section Référence des AWS CLI commandes.

`create-sample-findings`

L'exemple de code suivant montre comment utiliser`create-sample-findings`.

AWS CLI

Pour créer des échantillons de GuardDuty résultats dans la région actuelle.

Cet exemple montre comment créer un exemple de recherche parmi les types fournis.

```
aws guardduty create-sample-findings \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Exemples de résultats dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateSampleFindingsà la section Référence des AWS CLI commandes.

`create-threat-intel-set`

L'exemple de code suivant montre comment utiliser`create-threat-intel-set`.

AWS CLI

Pour créer et activer un nouvel ensemble d'informations sur les menaces

L'`create-threat-intel-set`exemple suivant crée et active un ensemble d'informations sur les menaces dans la région actuelle.

```
aws guardduty create-threat-intel-set \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myThreatSet-example \
    --format TXT \
    --location s3://amzn-s3-demo-bucket/threatlist.csv \
    --activate
```

Sortie :

```
{
    "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"
}
```

Pour plus d'informations, consultez la section Utilisation de listes d'adresses IP fiables et de listes de menaces dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateThreatIntelSetà la section Référence des AWS CLI commandes.

`decline-invitations`

L'exemple de code suivant montre comment utiliser`decline-invitations`.

AWS CLI

Pour refuser une invitation à faire gérer Guardduty par un autre compte dans la région actuelle.

Cet exemple montre comment refuser une invitation d'adhésion.

```
aws guardduty decline-invitations \
    --account-ids 111122223333
```

Sortie :

```
{
    "UnprocessedAccounts": []
}
```

Pour plus d'informations, consultez la section Gestion GuardDuty des comptes sur invitation dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeclineInvitationsà la section Référence des AWS CLI commandes.

`delete-detector`

L'exemple de code suivant montre comment utiliser`delete-detector`.

AWS CLI

Pour supprimer un détecteur et le désactiver GuardDuty dans la région actuelle.

Cet exemple montre comment supprimer un détecteur. En cas de succès, cela sera désactivé GuardDuty dans la région associée à ce détecteur.

```
aws guardduty delete-detector \
    --detector-id b6b992d6d2f48e64bc59180bfexample
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suspension ou désactivation GuardDuty dans le guide de l'GuardDuty utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteDetectorà la section Référence des AWS CLI commandes.

`delete-filter`

L'exemple de code suivant montre comment utiliser`delete-filter`.

AWS CLI

Pour supprimer un filtre existant dans la région actuelle

Cet exemple montre comment créer et supprimer un filtre.

```
aws guardduty delete-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --filter-name byebyeFilter
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Résultats du filtrage dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteFilterà la section Référence des AWS CLI commandes.

`disable-organization-admin-account`

L'exemple de code suivant montre comment utiliser`disable-organization-admin-account`.

AWS CLI

Pour supprimer un compte en tant qu'administrateur délégué au GuardDuty sein de votre organisation

Cet exemple montre comment supprimer un compte en tant qu'administrateur délégué pour GuardDuty.

```
aws guardduty disable-organization-admin-account \
    --admin-account-id 111122223333
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes auprès AWS des organisations dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DisableOrganizationAdminAccountà la section Référence des AWS CLI commandes.

`disassociate-from-master-account`

L'exemple de code suivant montre comment utiliser`disassociate-from-master-account`.

AWS CLI

Pour vous dissocier de votre compte administrateur actuel dans la région actuelle

L'`disassociate-from-master-account`exemple suivant dissocie votre compte du compte GuardDuty administrateur actuel dans la AWS région actuelle.

```
aws guardduty disassociate-from-master-account \
    --detector-id d4b040365221be2b54a6264dcexample
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Comprendre la relation entre le compte GuardDuty administrateur et les comptes membres dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DisassociateFromMasterAccountà la section Référence des AWS CLI commandes.

`get-detector`

L'exemple de code suivant montre comment utiliser`get-detector`.

AWS CLI

Pour récupérer les détails d'un détecteur spécifique

L'`get-detector`exemple suivant affiche les détails de configuration du détecteur spécifié.

```
aws guardduty get-detector \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

Sortie :

```
{
    "Status": "ENABLED",
    "ServiceRole": "arn:aws:iam::111122223333:role/aws-service-role/
guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDuty",
    "Tags": {},
    "FindingPublishingFrequency": "SIX_HOURS",
    "UpdatedAt": "2018-11-07T03:24:22.938Z",
    "CreatedAt": "2017-12-22T22:51:31.940Z"
}
```

Pour plus d'informations, consultez la section Concepts et terminologie du guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetDetectorà la section Référence des AWS CLI commandes.

`get-findings`

L'exemple de code suivant montre comment utiliser`get-findings`.

AWS CLI

Exemple 1 : Pour récupérer les détails d'un résultat spécifique

L'`get-findings`exemple suivant récupère les détails complets de la recherche JSON de la recherche spécifiée.

```
aws guardduty get-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-id 1ab92989eaf0e742df4a014d5example
```

Sortie :

```
{
    "Findings": [
        {
            "Resource": {
                "ResourceType": "AccessKey",
                "AccessKeyDetails": {
                    "UserName": "testuser",
```

```
                    "UserType": "IAMUser",
                    "PrincipalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "AccessKeyId": "ASIASZ4SI7REEEXAMPLE"
                }
            },
            "Description": "APIs commonly used to discover the users, groups,
policies and permissions in an account, was invoked by IAM principal testuser under
unusual circumstances. Such activity is not typically seen from this principal.",
            "Service": {
                "Count": 5,
                "Archived": false,
                "ServiceName": "guardduty",
                "EventFirstSeen": "2020-05-26T22:02:24Z",
                "ResourceRole": "TARGET",
                "EventLastSeen": "2020-05-26T22:33:55Z",
                "DetectorId": "d4b040365221be2b54a6264dcexample",
                "Action": {
                    "ActionType": "AWS_API_CALL",
                    "AwsApiCallAction": {
                        "RemoteIpDetails": {
                            "GeoLocation": {
                                "Lat": 51.5164,
                                "Lon": -0.093
                            },
                            "City": {
                                "CityName": "London"
                            },
                            "IpAddressV4": "52.94.36.7",
                            "Organization": {
                                "Org": "Amazon.com",
                                "Isp": "Amazon.com",
                                "Asn": "16509",
                                "AsnOrg": "AMAZON-02"
                            },
                            "Country": {
                                "CountryName": "United Kingdom"
                            }
                        },
                        "Api": "ListPolicyVersions",
                        "ServiceName": "iam.amazonaws.com",
                        "CallerType": "Remote IP"
                    }
                }
            },
```

```
            "Title": "Unusual user permission reconnaissance activity by testuser.",
            "Type": "Recon:IAMUser/UserPermissions",
            "Region": "us-east-1",
            "Partition": "aws",
            "Arn": "arn:aws:guardduty:us-east-1:111122223333:detector/
d4b040365221be2b54a6264dcexample/finding/1ab92989eaf0e742df4a014d5example",
            "UpdatedAt": "2020-05-26T22:55:21.703Z",
            "SchemaVersion": "2.0",
            "Severity": 5,
            "Id": "1ab92989eaf0e742df4a014d5example",
            "CreatedAt": "2020-05-26T22:21:48.385Z",
            "AccountId": "111122223333"
        }
    ]
}
```

Pour plus d'informations, consultez les résultats du guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetFindingsà la section Référence des AWS CLI commandes.

`get-ip-set`

L'exemple de code suivant montre comment utiliser`get-ip-set`.

AWS CLI

Pour répertorier, obtenir des informations sur un ensemble d'adresses IP fiables spécifié

L'`get-ip-set`exemple suivant montre le statut et les détails de l'ensemble d'adresses IP fiables spécifié.

```
aws guardduty get-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --ip-set-id d4b94fc952d6912b8f3060768example
```

Sortie :

```
{
    "Status": "ACTIVE",
    "Location": "s3://amzn-s3-demo-bucket.s3-us-west-2.amazonaws.com/
customlist.csv",
```

```
    "Tags": {},
    "Format": "TXT",
    "Name": "test-ip-set-example"
}
```

Pour plus d'informations, consultez la section Utilisation de listes d'adresses IP fiables et de listes de menaces dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetIpSetà la section Référence des AWS CLI commandes.

`get-master-account`

L'exemple de code suivant montre comment utiliser`get-master-account`.

AWS CLI

Pour récupérer les informations relatives à votre compte principal dans la région actuelle

L'`get-master-account`exemple suivant affiche le statut et les détails du compte principal associé à votre détecteur dans la région actuelle.

```
aws guardduty get-master-account \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

Sortie :

```
{
    "Master": {
        "InvitationId": "04b94d9704854a73f94e061e8example",
        "InvitedAt": "2020-06-09T22:23:04.970Z",
        "RelationshipStatus": "Enabled",
        "AccountId": "111122223333"
    }
}
```

Pour plus d'informations, voir Comprendre la relation entre le compte GuardDuty administrateur et le compte membre dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetMasterAccountà la section Référence des AWS CLI commandes.

`list-detectors`

L'exemple de code suivant montre comment utiliser`list-detectors`.

AWS CLI

Pour répertorier les détecteurs disponibles dans la région actuelle

L'`list-detectors`exemple suivant répertorie les détecteurs disponibles dans votre AWS région actuelle.

```
aws guardduty list-detectors
```

Sortie :

```
{
    "DetectorIds": [
        "12abc34d567e8fa901bc2d34eexample"
    ]
}
```

Pour plus d'informations, consultez la section Concepts et terminologie du guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListDetectorsà la section Référence des AWS CLI commandes.

`list-findings`

L'exemple de code suivant montre comment utiliser`list-findings`.

AWS CLI

Exemple 1 : pour répertorier tous les résultats de la région actuelle

L'`list-findings`exemple suivant affiche une liste de tous les FindingID pour la région actuelle, triés par gravité du plus élevé au plus faible.

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
```

```
    --sort-criteria '{"AttributeName": "severity","OrderBy":"DESC"}'
```

Sortie :

```
{
    "FindingIds": [
        "04b8ab50fd29c64fc771b232dexample",
        "5ab8ab50fd21373735c826d3aexample",
        "90b93de7aba69107f05bbe60bexample",
        ...
    ]
}
```

Pour plus d'informations, consultez les résultats du guide de GuardDuty l'utilisateur.

Exemple 2 : Pour répertorier les résultats de la région actuelle correspondant à un critère de recherche spécifique

L'`list-findings`exemple suivant affiche une liste de tous les FindingID correspondant à un type de recherche spécifié.

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-criteria  '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/
SSHBruteForce"]}}}'
```

Sortie :

```
{
    "FindingIds": [
        "90b93de7aba69107f05bbe60bexample",
        "6eb9430d7023d30774d6f05e3example",
        "2eb91a2d060ac9a21963a5848example",
        "44b8ab50fd2b0039a9e48f570example",
        "9eb8ab4cd2b7e5b66ba4f5e96example",
        "e0b8ab3a38e9b0312cc390ceeexample"
    ]
}
```

Pour plus d'informations, consultez les résultats du guide de GuardDuty l'utilisateur.

Exemple 3 : pour répertorier les résultats de la région actuelle correspondant à un ensemble spécifique de critères de recherche définis dans un fichier JSON

L'`list-findings`exemple suivant affiche une liste de tous les FindingID qui ne sont pas archivés et impliquent l'utilisateur IAM nommé « testuser », tel que spécifié dans un fichier JSON.

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-criteria  file://myfile.json
```

Contenu de `myfile.json` :

```
{"Criterion": {
    "resource.accessKeyDetails.userName":{
                "Eq":[
                    "testuser"
                    ]
                },
    "service.archived": {
                "Eq": [
                    "false"
                ]
            }
        }
}
```

Sortie :

```
{
    "FindingIds": [
        "1ab92989eaf0e742df4a014d5example"
    ]
}
```

Pour plus d'informations, consultez les résultats du guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListFindingsà la section Référence des AWS CLI commandes.

`list-invitations`

L'exemple de code suivant montre comment utiliser`list-invitations`.

AWS CLI

Pour répertorier les détails de vos invitations à devenir membre dans la région actuelle

L'`list-invitations`exemple suivant répertorie les détails et les statuts de vos invitations à devenir GuardDuty membre dans la région actuelle.

```
aws guardduty list-invitations
```

Sortie :

```
{
    "Invitations": [
        {
            "InvitationId": "d6b94fb03a66ff665f7db8764example",
            "InvitedAt": "2020-06-10T17:56:38.221Z",
            "RelationshipStatus": "Invited",
            "AccountId": "123456789111"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion GuardDuty des comptes sur invitation dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListInvitationsà la section Référence des AWS CLI commandes.

`list-ip-sets`

L'exemple de code suivant montre comment utiliser`list-ip-sets`.

AWS CLI

Pour répertorier les ensembles d'adresses IP fiables dans la région actuelle

L'`list-ip-sets`exemple suivant répertorie les ensembles d'adresses IP fiables de votre AWS région actuelle.

```
aws guardduty list-ip-sets \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

Sortie :

```
{
    "IpSetIds": [
        "d4b94fc952d6912b8f3060768example"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de listes d'adresses IP fiables et de listes de menaces dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListIpSetsà la section Référence des AWS CLI commandes.

`list-members`

L'exemple de code suivant montre comment utiliser`list-members`.

AWS CLI

Exemple 1 : pour répertorier uniquement les membres actuels de la région actuelle

L'`list-members`exemple suivant répertorie et fournit des détails uniquement sur les comptes de membres actuels associés au compte d' GuardDuty administrateur, dans la région actuelle.

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --only-associated="true"
```

Sortie :

```
{
    "Members": [
        {
            "RelationshipStatus": "Enabled",
            "InvitedAt": "2020-06-09T22:49:00.910Z",
            "MasterId": "111122223333",
            "DetectorId": "7ab8b2f61b256c87f793f6a86example",
            "UpdatedAt": "2020-06-09T23:08:22.512Z",
            "Email": "your+member@example.com",
            "AccountId": "123456789012"
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Comprendre la relation entre le compte GuardDuty administrateur et les comptes membres dans le Guide de GuardDuty l'utilisateur.

Exemple 2 : pour répertorier tous les membres de la région actuelle

L'`list-members`exemple suivant répertorie et fournit des détails sur tous les comptes de membres, y compris ceux qui ont été dissociés ou qui n'ont pas encore accepté l'invitation de l' GuardDuty administrateur, dans la région actuelle.

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --only-associated="false"
```

Sortie :

```
{
    "Members": [
        {
            "RelationshipStatus": "Enabled",
            "InvitedAt": "2020-06-09T22:49:00.910Z",
            "MasterId": "111122223333",
            "DetectorId": "7ab8b2f61b256c87f793f6a86example",
            "UpdatedAt": "2020-06-09T23:08:22.512Z",
            "Email": "your+other+member@example.com",
            "AccountId": "555555555555"
        }
    ]
}
```

Pour plus d'informations, consultez la section Comprendre la relation entre le compte GuardDuty administrateur et les comptes membres dans le Guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListMembersà la section Référence des AWS CLI commandes.

`update-ip-set`

L'exemple de code suivant montre comment utiliser`update-ip-set`.

AWS CLI

Pour mettre à jour un ensemble d'adresses IP fiables

L'`update-ip-set`exemple suivant montre comment mettre à jour les détails d'un ensemble d'adresses IP fiables.

```
aws guardduty update-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --ip-set-id d4b94fc952d6912b8f3060768example \
    --location https://amzn-s3-demo-bucket.s3-us-west-2.amazonaws.com/
customtrustlist2.csv
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de listes d'adresses IP fiables et de listes de menaces dans le guide de GuardDuty l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateIpSetà la section Référence des AWS CLI commandes.

AWS Health exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Health.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-affected-entities`

L'exemple de code suivant montre comment utiliser`describe-affected-entities`.

AWS CLI

Pour répertorier les entités touchées par un événement de AWS santé spécifique

L'`describe-affected-entities`exemple suivant répertorie les entités affectées par l'événement AWS Health spécifié. Cet événement est une notification de facturation pour le AWS compte.

```
aws health describe-affected-entities \
    --filter "eventArns=arn:aws:health:global::event/BILLING/
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-
EXAMPLE11145" \
    --region us-east-1
```

Sortie :

```
{
    "entities": [
        {
            "entityArn": "arn:aws:health:global:123456789012:entity/
EXAMPLEimSMoULmWHpb",
            "eventArn": "arn:aws:health:global::event/BILLING/
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-
EXAMPLE11145",
            "entityValue": "AWS_ACCOUNT",
            "awsAccountId": "123456789012",
            "lastUpdatedTime": 1588356454.08
        }
    ]
}
```

Pour plus d'informations, consultez la section Journal des événements dans le AWS Health User Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeAffectedEntitiesà la section Référence des AWS CLI commandes.

`describe-event-details`

L'exemple de code suivant montre comment utiliser`describe-event-details`.

AWS CLI

Pour répertorier les informations relatives à un événement AWS de santé

L'`describe-event-details`exemple suivant répertorie les informations relatives à l'événement AWS Health spécifié.

```
aws health describe-event-details \
    --event-arns "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111" \
    --region us-east-1
```

Sortie :

```
{
    "successfulSet": [
        {
            "event": {
                "arn": "arn:aws:health:us-east-1::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
                "service": "EC2",
                "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
                "eventTypeCategory": "issue",
                "region": "us-east-1",
                "startTime": 1587462325.096,
                "endTime": 1587464204.774,
                "lastUpdatedTime": 1587464204.865,
                "statusCode": "closed"
            },
            "eventDescription": {
                "latestDescription": "[RESOLVED] Increased API Error Rates and
 Latencies\n\n[02:45 AM PDT] We are investigating increased API error rates and
 latencies in the US-EAST-1 Region.\n\n[03:16 AM PDT] Between 2:10 AM and 2:59 AM
 PDT we experienced increased API error rates and latencies in the US-EAST-1 Region.
 The issue has been resolved and the service is operating normally."
            }
        }
    ],
    "failedSet": []
}
```

Pour plus d'informations, consultez le volet Détails de l'événement dans le AWS Health User Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEventDetailsà la section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Exemple 1 : Pour répertorier les événements liés à AWS la santé

L'`describe-events`exemple suivant répertorie les récents événements liés à AWS la santé.

```
aws health describe-events \
    --region us-east-1
```

Sortie :

```
{
    "events": [
        {
            "arn": "arn:aws:health:us-west-1::event/ECS/AWS_ECS_OPERATIONAL_ISSUE/
AWS_ECS_OPERATIONAL_ISSUE_KWQPY_EXAMPLE111",
            "service": "ECS",
            "eventTypeCode": "AWS_ECS_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-west-1",
            "startTime": 1589077890.53,
            "endTime": 1589086345.597,
            "lastUpdatedTime": 1589086345.905,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/BILLING/AWS_BILLING_NOTIFICATION/
AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-EXAMPLE1118b",
            "service": "BILLING",
            "eventTypeCode": "AWS_BILLING_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1588356000.0,
            "lastUpdatedTime": 1588356524.358,
            "statusCode": "open",
```

```
            "eventScopeCode": "ACCOUNT_SPECIFIC"
        },
        {
            "arn": "arn:aws:health:us-west-2::event/
CLOUDFORMATION/AWS_CLOUDFORMATION_OPERATIONAL_ISSUE/
AWS_CLOUDFORMATION_OPERATIONAL_ISSUE_OHTWY_EXAMPLE111",
            "service": "CLOUDFORMATION",
            "eventTypeCode": "AWS_CLOUDFORMATION_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-west-2",
            "startTime": 1588279630.761,
            "endTime": 1588284650.0,
            "lastUpdatedTime": 1588284691.941,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:ap-northeast-1::event/LAMBDA/
AWS_LAMBDA_OPERATIONAL_ISSUE/AWS_LAMBDA_OPERATIONAL_ISSUE_JZDND_EXAMPLE111",
            "service": "LAMBDA",
            "eventTypeCode": "AWS_LAMBDA_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-northeast-1",
            "startTime": 1587379534.08,
            "endTime": 1587391771.0,
            "lastUpdatedTime": 1587395689.316,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1586473044.284,
            "endTime": 1586479706.091,
            "lastUpdatedTime": 1586479706.153,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
```

```
            "arn": "arn:aws:health:global::event/SECURITY/AWS_SECURITY_NOTIFICATION/
AWS_SECURITY_NOTIFICATION_42007387-8129-42da-8c88-EXAMPLE11139",
            "service": "SECURITY",
            "eventTypeCode": "AWS_SECURITY_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1585674000.0,
            "lastUpdatedTime": 1585674004.132,
            "statusCode": "open",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/CLOUDFRONT/
AWS_CLOUDFRONT_OPERATIONAL_ISSUE/AWS_CLOUDFRONT_OPERATIONAL_ISSUE_FRQXG_EXAMPLE111",
            "service": "CLOUDFRONT",
            "eventTypeCode": "AWS_CLOUDFRONT_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "global",
            "startTime": 1585610898.589,
            "endTime": 1585617671.0,
            "lastUpdatedTime": 1585620638.869,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/SES/AWS_SES_OPERATIONAL_ISSUE/
AWS_SES_OPERATIONAL_ISSUE_URNDF_EXAMPLE111",
            "service": "SES",
            "eventTypeCode": "AWS_SES_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1585342008.46,
            "endTime": 1585344017.0,
            "lastUpdatedTime": 1585344355.989,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/IAM/
AWS_IAM_OPERATIONAL_NOTIFICATION/
AWS_IAM_OPERATIONAL_NOTIFICATION_b6771c34-6ecd-4aea-9d3e-EXAMPLE1117e",
            "service": "IAM",
            "eventTypeCode": "AWS_IAM_OPERATIONAL_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
```

```
            "region": "global",
            "startTime": 1584978300.0,
            "lastUpdatedTime": 1584978553.572,
            "statusCode": "open",
            "eventScopeCode": "ACCOUNT_SPECIFIC"
        },
        {
            "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-southeast-2",
            "startTime": 1583881487.483,
            "endTime": 1583885056.785,
            "lastUpdatedTime": 1583885057.052,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with the AWS Personal Health Dashboard dans le AWS Health User Guide.

Exemple 2 : Pour répertorier les événements AWS de santé par service et code d'état de l'événement

L'`describe-events`exemple suivant répertorie les événements AWS Health pour Amazon Elastic Compute Cloud (Amazon EC2) dont le statut est fermé.

```
aws health describe-events \
    --filter "services=EC2,eventStatusCodes=closed"
```

Sortie :

```
{
    "events": [
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
            "service": "EC2",
```

```
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1587462325.096,
            "endTime": 1587464204.774,
            "lastUpdatedTime": 1587464204.865,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1586473044.284,
            "endTime": 1586479706.091,
            "lastUpdatedTime": 1586479706.153,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-southeast-2",
            "startTime": 1583881487.483,
            "endTime": 1583885056.785,
            "lastUpdatedTime": 1583885057.052,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with the AWS Personal Health Dashboard dans le AWS Health User Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeEventsà la section Référence des AWS CLI commandes.

HealthImaging exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with HealthImaging.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`copy-image-set`

L'exemple de code suivant montre comment utiliser`copy-image-set`.

AWS CLI

Exemple 1 : pour copier un ensemble d'images sans destination.

L'`copy-image-set`exemple suivant crée une copie d'une série d'images sans destination.

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

Sortie :

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042357.432,
```

```
        "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042357.432,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
        "createdAt": 1680027126.436
    },
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 2 : pour copier une série d'images avec une destination.

L'`copy-image-set`exemple suivant crée une copie d'une série d'images avec une destination.

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" },
 "destinationImageSet": { "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
 "latestVersionId": "1"} }'
```

Sortie :

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042505.135,
        "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042505.135,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
```

```
        "createdAt": 1680027126.436
    },
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 3 : copier un sous-ensemble d'instances d'un jeu d'images source vers un ensemble d'images de destination.

L'`copy-image-set`exemple suivant copie une instance DICOM du jeu d'images source vers le jeu d'images de destination. Le paramètre de force est fourni pour corriger les incohérences dans les attributs des niveaux Patient, Study et Series.

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId":
 "1","DICOMCopies": {"copiableAttributes": "{\"SchemaVersion\":\"1.1\",\"Study\":
{\"Series\":{\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3666.0\":
{\"Instances\":
{\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3669.0\":
{}}}}}}"}},"destinationImageSet": {"imageSetId":
 "b9eb50d8ee682eb9fcf4acbf92f62bb7","latestVersionId": "1"}}' \
    --force
```

Sortie :

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042505.135,
        "imageSetId": "b9eb50d8ee682eb9fcf4acbf92f62bb7",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042505.135,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
        "createdAt": 1680027126.436
```

```
    },
    "datastoreId": "12345678901234567890123456789012"
}
```

Pour plus d'informations, consultez la section Copier un ensemble d'images dans le manuel du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous CopyImageSetà la section Référence des AWS CLI commandes.

`create-datastore`

L'exemple de code suivant montre comment utiliser`create-datastore`.

AWS CLI

Pour créer un magasin de données

L'exemple de `create-datastore` code suivant crée un magasin de données portant le nom`my-datastore`.

```
aws medical-imaging create-datastore \
    --datastore-name "my-datastore"
```

Sortie :

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "CREATING"
}
```

Pour plus d'informations, consultez la section Création d'un magasin de données dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous CreateDatastoreà la section Référence des AWS CLI commandes.

`delete-datastore`

L'exemple de code suivant montre comment utiliser`delete-datastore`.

AWS CLI

Pour supprimer un magasin de données

L'exemple de `delete-datastore` code suivant supprime un magasin de données.

```
aws medical-imaging delete-datastore \
    --datastore-id "12345678901234567890123456789012"
```

Sortie :

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "DELETING"
}
```

Pour plus d'informations, consultez la section Suppression d'un magasin de données dans le guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous DeleteDatastoreà la section Référence des AWS CLI commandes.

`delete-image-set`

L'exemple de code suivant montre comment utiliser`delete-image-set`.

AWS CLI

Pour supprimer un ensemble d'images

L'exemple de `delete-image-set` code suivant supprime un ensemble d'images.

```
aws medical-imaging delete-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Sortie :

```
{
    "imageSetWorkflowStatus": "DELETING",
```

```
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "datastoreId": "12345678901234567890123456789012"
}
```

Pour plus d'informations, consultez la section Suppression d'un ensemble d'images dans le manuel du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous DeleteImageSetà la section Référence des AWS CLI commandes.

`get-datastore`

L'exemple de code suivant montre comment utiliser`get-datastore`.

AWS CLI

Pour obtenir les propriétés d'un magasin de données

L'exemple de `get-datastore` code suivant permet d'obtenir les propriétés d'un magasin de données.

```
aws medical-imaging get-datastore \
    --datastore-id 12345678901234567890123456789012
```

Sortie :

```
{
    "datastoreProperties": {
        "datastoreId": "12345678901234567890123456789012",
        "datastoreName": "TestDatastore123",
        "datastoreStatus": "ACTIVE",
        "datastoreArn": "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012",
        "createdAt": "2022-11-15T23:33:09.643000+00:00",
        "updatedAt": "2022-11-15T23:33:09.643000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Obtenir les propriétés du magasin de données dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous GetDatastoreà la section Référence des AWS CLI commandes.

`get-dicom-import-job`

L'exemple de code suivant montre comment utiliser`get-dicom-import-job`.

AWS CLI

Pour obtenir les propriétés d'une tâche d'importation DICOM

L'exemple de `get-dicom-import-job` code suivant permet d'obtenir les propriétés d'une tâche d'importation DICOM.

```
aws medical-imaging get-dicom-import-job \
    --datastore-id "12345678901234567890123456789012" \
    --job-id "09876543210987654321098765432109"
```

Sortie :

```
{
    "jobProperties": {
        "jobId": "09876543210987654321098765432109",
        "jobName": "my-job",
        "jobStatus": "COMPLETED",
        "datastoreId": "12345678901234567890123456789012",
        "dataAccessRoleArn": "arn:aws:iam::123456789012:role/
ImportJobDataAccessRole",
        "endedAt": "2022-08-12T11:29:42.285000+00:00",
        "submittedAt": "2022-08-12T11:28:11.152000+00:00",
        "inputS3Uri": "s3://medical-imaging-dicom-input/dicom_input/",
        "outputS3Uri": "s3://medical-imaging-output/
job_output/12345678901234567890123456789012-
DicomImport-09876543210987654321098765432109/"
    }
}
```

Pour plus d'informations, consultez la section Obtenir les propriétés des tâches d'importation dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, voir Get DICOMImport Job in AWS CLI Command Reference.

`get-image-frame`

L'exemple de code suivant montre comment utiliser`get-image-frame`.

AWS CLI

Pour obtenir les données relatives aux pixels définis par image

L'exemple de `get-image-frame` code suivant permet d'obtenir un cadre d'image.

```
aws medical-imaging get-image-frame \
    --datastore-id "12345678901234567890123456789012" \
    --image-set-id "98765412345612345678907890789012" \
    --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \
    imageframe.jph
```

Remarque : Cet exemple de code n'inclut pas de sortie car l' GetImageFrame action renvoie un flux de données de pixels vers le fichier imageframe.jph. Pour plus d'informations sur le décodage et l'affichage de trames d'images, consultez la section Bibliothèques de décodage HTJ2 K.

Pour plus d'informations, consultez la section Obtenir des données en pixels d'un ensemble d'images dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous GetImageFrameà la section Référence des AWS CLI commandes.

`get-image-set-metadata`

L'exemple de code suivant montre comment utiliser`get-image-set-metadata`.

AWS CLI

Exemple 1 : Pour obtenir les métadonnées d'un ensemble d'images sans version

L'exemple de `get-image-set-metadata` code suivant permet d'obtenir les métadonnées d'un ensemble d'images sans spécifier de version.

Remarque : `outfile` est un paramètre obligatoire

```
aws medical-imaging get-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
```

```
    studymetadata.json.gz
```

Les métadonnées renvoyées sont compressées avec gzip et stockées dans le fichier studymetadata.json.gz. Pour visualiser le contenu de l'objet JSON renvoyé, vous devez d'abord le décompresser.

Sortie :

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

Exemple 2 : Pour obtenir les métadonnées d'un ensemble d'images avec la version

L'exemple de `get-image-set-metadata` code suivant permet d'obtenir les métadonnées d'un ensemble d'images avec une version spécifiée.

Remarque : `outfile` est un paramètre obligatoire

```
aws medical-imaging get-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --version-id 1 \
    studymetadata.json.gz
```

Les métadonnées renvoyées sont compressées avec gzip et stockées dans le fichier studymetadata.json.gz. Pour visualiser le contenu de l'objet JSON renvoyé, vous devez d'abord le décompresser.

Sortie :

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

Pour plus d'informations, consultez la section Obtenir les métadonnées d'un ensemble d'images dans le manuel du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous GetImageSetMetadataà la section Référence des AWS CLI commandes.

`get-image-set`

L'exemple de code suivant montre comment utiliser`get-image-set`.

AWS CLI

Pour obtenir les propriétés des ensembles d'images

L'exemple de `get-image-set` code suivant permet d'obtenir les propriétés d'un ensemble d'images.

```
aws medical-imaging get-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 18f88ac7870584f58d56256646b4d92b \
    --version-id 1
```

Sortie :

```
{
    "versionId": "1",
    "imageSetWorkflowStatus": "COPIED",
    "updatedAt": 1680027253.471,
    "imageSetId": "18f88ac7870584f58d56256646b4d92b",
    "imageSetState": "ACTIVE",
    "createdAt": 1679592510.753,
    "datastoreId": "12345678901234567890123456789012"
}
```

Pour plus d'informations, consultez la section Obtenir les propriétés d'un ensemble d'images dans le manuel du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous GetImageSetà la section Référence des AWS CLI commandes.

`list-datastores`

L'exemple de code suivant montre comment utiliser`list-datastores`.

AWS CLI

Pour répertorier les magasins de données

L'exemple de `list-datastores` code suivant répertorie les magasins de données disponibles.

```
aws medical-imaging list-datastores
```

Sortie :

```
{
    "datastoreSummaries": [
        {
            "datastoreId": "12345678901234567890123456789012",
            "datastoreName": "TestDatastore123",
            "datastoreStatus": "ACTIVE",
            "datastoreArn": "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012",
            "createdAt": "2022-11-15T23:33:09.643000+00:00",
            "updatedAt": "2022-11-15T23:33:09.643000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les banques de données dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous ListDatastoresà la section Référence des AWS CLI commandes.

`list-dicom-import-jobs`

L'exemple de code suivant montre comment utiliser`list-dicom-import-jobs`.

AWS CLI

Pour répertorier les tâches d'importation Dicom

L'exemple de `list-dicom-import-jobs` code suivant répertorie les tâches d'importation dicom.

```
aws medical-imaging list-dicom-import-jobs \
    --datastore-id "12345678901234567890123456789012"
```

Sortie :

```
{
    "jobSummaries": [
        {
            "jobId": "09876543210987654321098765432109",
            "jobName": "my-job",
            "jobStatus": "COMPLETED",
            "datastoreId": "12345678901234567890123456789012",
            "dataAccessRoleArn": "arn:aws:iam::123456789012:role/
ImportJobDataAccessRole",
            "endedAt": "2022-08-12T11:21:56.504000+00:00",
            "submittedAt": "2022-08-12T11:20:21.734000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les tâches d'importation dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, consultez la section DICOMImportRépertorier les tâches dans AWS CLI Command Reference.

`list-image-set-versions`

L'exemple de code suivant montre comment utiliser`list-image-set-versions`.

AWS CLI

Pour répertorier les versions d'ensembles d'images

L'exemple de `list-image-set-versions` code suivant répertorie l'historique des versions d'un ensemble d'images.

```
aws medical-imaging list-image-set-versions \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Sortie :

```
{
    "imageSetPropertiesList": [
        {
            "ImageSetWorkflowStatus": "UPDATED",
```

```
            "versionId": "4",
            "updatedAt": 1680029436.304,
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "createdAt": 1680027126.436
        },
        {
            "ImageSetWorkflowStatus": "UPDATED",
            "versionId": "3",
            "updatedAt": 1680029163.325,
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "createdAt": 1680027126.436
        },
        {
            "ImageSetWorkflowStatus": "COPY_FAILED",
            "versionId": "2",
            "updatedAt": 1680027455.944,
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "message": "INVALID_REQUEST:  Series of SourceImageSet and
 DestinationImageSet don't match.",
            "createdAt": 1680027126.436
        },
        {
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "versionId": "1",
            "ImageSetWorkflowStatus": "COPIED",
            "createdAt": 1680027126.436
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les versions des ensembles d'images dans le manuel du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous ListImageSetVersionsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Exemple 1 : pour répertorier les balises de ressources d'un magasin de données

L'exemple de `list-tags-for-resource` code suivant répertorie les balises d'un magasin de données.

```
aws medical-imaging list-tags-for-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012"
```

Sortie :

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

Exemple 2 : pour répertorier les balises de ressources pour un ensemble d'images

L'exemple de `list-tags-for-resource` code suivant répertorie les balises d'un ensemble d'images.

```
aws medical-imaging list-tags-for-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b"
```

Sortie :

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

Pour plus d'informations, consultez la section Marquage des ressources avec AWS HealthImaging dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`search-image-sets`

L'exemple de code suivant montre comment utiliser`search-image-sets`.

AWS CLI

Exemple 1 : Pour rechercher des ensembles d'images avec un opérateur EQUAL

L'exemple de `search-image-sets` code suivant utilise l'opérateur EQUAL pour rechercher des ensembles d'images en fonction d'une valeur spécifique.

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria file://search-criteria.json
```

Contenu de `search-criteria.json`

```
{
    "filters": [{
        "values": [{"DICOMPatientId" : "SUBJECT08701"}],
        "operator": "EQUAL"
    }]
}
```

Sortie :

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
             "DICOMPatientSex": "F",
             "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
             "DICOMPatientBirthDate": "19201120",
             "DICOMStudyDescription": "UNKNOWN",
             "DICOMPatientId": "SUBJECT08701",
             "DICOMPatientName": "Melissa844 Huel628",
             "DICOMNumberOfStudyRelatedInstances": 1,
             "DICOMStudyTime": "140728",
```

```
             "DICOMNumberOfStudyRelatedSeries": 1
            },
        "updatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

Exemple 2 : Pour rechercher des ensembles d'images à l'aide de l'opérateur BETWEEN en utilisant DICOMStudy la date et l' DICOMStudyheure

L'exemple de `search-image-sets` code suivant recherche des ensembles d'images contenant des études DICOM générées entre le 1er janvier 1990 (00h00) et le 1er janvier 2023 (00h00).

Remarque : DICOMStudy L'heure est facultative. S'il n'est pas présent, 00 h 00 (début de journée) est la valeur horaire pour les dates fournies pour le filtrage.

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria file://search-criteria.json
```

Contenu de `search-criteria.json`

```
{
    "filters": [{
        "values": [{
            "DICOMStudyDateAndTime": {
                "DICOMStudyDate": "19900101",
                "DICOMStudyTime": "000000"
            }
        },
        {
            "DICOMStudyDateAndTime": {
                "DICOMStudyDate": "20230101",
                "DICOMStudyTime": "000000"
            }
        }],
        "operator": "BETWEEN"
    }]
}
```

Sortie :

```
{
```

```
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
            "DICOMPatientSex": "F",
            "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
            "DICOMPatientBirthDate": "19201120",
            "DICOMStudyDescription": "UNKNOWN",
            "DICOMPatientId": "SUBJECT08701",
            "DICOMPatientName": "Melissa844 Huel628",
            "DICOMNumberOfStudyRelatedInstances": 1,
            "DICOMStudyTime": "140728",
            "DICOMNumberOfStudyRelatedSeries": 1
        },
        "updatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

Exemple 3 : Pour rechercher des ensembles d'images avec un opérateur BETWEEN à l'aide de CreatedAt (les études temporelles étaient précédemment conservées)

L'exemple de `search-image-sets` code suivant recherche des ensembles d'images contenant des études DICOM persistantes HealthImaging entre les plages horaires du fuseau horaire UTC.

Remarque : Indiquez CreatedAt dans le format d'exemple (« 1985-04-12T 23:20:50,52 Z »).

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

Contenu de `search-criteria.json`

```
{
    "filters": [{
        "values": [{
            "createdAt": "1985-04-12T23:20:50.52Z"
        },
        {
            "createdAt": "2022-04-12T23:20:50.52Z"
```

```
        }],
        "operator": "BETWEEN"
    }]
}
```

Sortie :

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
            "DICOMPatientSex": "F",
            "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
            "DICOMPatientBirthDate": "19201120",
            "DICOMStudyDescription": "UNKNOWN",
            "DICOMPatientId": "SUBJECT08701",
            "DICOMPatientName": "Melissa844 Huel628",
            "DICOMNumberOfStudyRelatedInstances": 1,
            "DICOMStudyTime": "140728",
            "DICOMNumberOfStudyRelatedSeries": 1
        },
        "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

Exemple 4 : Pour rechercher des ensembles d'images avec un opérateur DICOMSeries EQUAL sur InstanceUid et BETWEEN sur UpdatedAt et trier les réponses dans l'ordre ASC sur le champ UpdatedAt

L'exemple de `search-image-sets` code suivant recherche des ensembles d'images avec un opérateur DICOMSeries EQUAL sur InstanceUid et BETWEEN sur UpdatedAt et trie la réponse dans l'ordre ASC sur le champ UpdatedAt.

Remarque : Fournissez UpdateDat dans le format d'exemple (« 1985-04-12T 23:20:50,52 Z »).

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

Contenu de `search-criteria.json`

```
{
    "filters": [{
        "values": [{
            "updatedAt": "2024-03-11T15:00:05.074000-07:00"
        }, {
            "updatedAt": "2024-03-11T16:00:05.074000-07:00"
        }],
        "operator": "BETWEEN"
    }, {
        "values": [{
            "DICOMSeriesInstanceUID": "1.2.840.99999999.84710745.943275268089"
        }],
        "operator": "EQUAL"
    }],
    "sort": {
        "sortField": "updatedAt",
        "sortOrder": "ASC"
    }
}
```

Sortie :

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
            "DICOMPatientSex": "F",
            "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
            "DICOMPatientBirthDate": "19201120",
            "DICOMStudyDescription": "UNKNOWN",
            "DICOMPatientId": "SUBJECT08701",
            "DICOMPatientName": "Melissa844 Huel628",
            "DICOMNumberOfStudyRelatedInstances": 1,
            "DICOMStudyTime": "140728",
            "DICOMNumberOfStudyRelatedSeries": 1
        },
        "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
```

```
    }]
}
```

Pour plus d'informations, consultez la section Recherche de séries d'images dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous SearchImageSetsà la section Référence des AWS CLI commandes.

`start-dicom-import-job`

L'exemple de code suivant montre comment utiliser`start-dicom-import-job`.

AWS CLI

Pour démarrer une tâche d'importation DICOM

L'exemple de `start-dicom-import-job` code suivant démarre une tâche d'importation DICOM.

```
aws medical-imaging start-dicom-import-job \
    --job-name "my-job" \
    --datastore-id "12345678901234567890123456789012" \
    --input-s3-uri "s3://medical-imaging-dicom-input/dicom_input/" \
    --output-s3-uri "s3://medical-imaging-output/job_output/" \
    --data-access-role-arn "arn:aws:iam::123456789012:role/ImportJobDataAccessRole"
```

Sortie :

```
{
    "datastoreId": "12345678901234567890123456789012",
    "jobId": "09876543210987654321098765432109",
    "jobStatus": "SUBMITTED",
    "submittedAt": "2022-08-12T11:28:11.152000+00:00"
}
```

Pour plus d'informations, consultez la section Démarrage d'une tâche d'importation dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, consultez Start DICOMImport Job dans AWS CLI Command Reference.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Exemple 1 : pour étiqueter un magasin de données

Les exemples de `tag-resource` code suivants balisent un magasin de données.

```
aws medical-imaging tag-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012" \
  --tags '{"Deployment":"Development"}'
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour baliser un ensemble d'images

Les exemples de `tag-resource` code suivants balisent un ensemble d'images.

```
aws medical-imaging tag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
    --tags '{"Deployment":"Development"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources avec AWS HealthImaging dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Exemple 1 : pour supprimer le balisage d'un magasin de données

L'exemple de `untag-resource` code suivant supprime les balises d'un magasin de données.

```
aws medical-imaging untag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012" \
    --tag-keys '["Deployment"]'
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour annuler le balisage d'un ensemble d'images

L'exemple de `untag-resource` code suivant supprime les balises d'un ensemble d'images.

```
aws medical-imaging untag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
    --tag-keys '["Deployment"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources avec AWS HealthImaging dans le Guide du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-image-set-metadata`

L'exemple de code suivant montre comment utiliser`update-image-set-metadata`.

AWS CLI

Exemple 1 : pour insérer ou mettre à jour un attribut dans les métadonnées d'un ensemble d'images

L'`update-image-set-metadata`exemple suivant insère ou met à jour un attribut dans les métadonnées du jeu d'images.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
```

```
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

Contenu de `metadata-updates.json`

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\":1.1,\"Patient\":{\"DICOM\":
{\"PatientName\":\"MX^MX\"}}}"
    }
}
```

Sortie :

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 2 : pour supprimer un attribut des métadonnées d'un ensemble d'images

L'`update-image-set-metadata`exemple suivant supprime un attribut des métadonnées du jeu d'images.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

Contenu de `metadata-updates.json`

```
{
```

```
    "DICOMUpdates": {
        "removableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"DICOM\":
{\"StudyDescription\":\"CHEST\"}}}"
    }
}
```

Sortie :

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 3 : pour supprimer une instance des métadonnées d'un ensemble d'images

L'`update-image-set-metadata`exemple suivant supprime une instance des métadonnées du jeu d'images.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

Contenu de `metadata-updates.json`

```
{
    "DICOMUpdates": {
        "removableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances\":
 {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {}}}}}}"
    }
}
```

Sortie :

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 4 : pour rétablir une version précédente d'un ensemble d'images

L'`update-image-set-metadata`exemple suivant montre comment rétablir une version antérieure d'un ensemble d'images. CopyImageSet et UpdateImageSetMetadata les actions créent de nouvelles versions de séries d'images.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 3 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates '{"revertToVersionId": "1"}'
```

Sortie :

```
{
    "datastoreId": "12345678901234567890123456789012",
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "latestVersionId": "4",
    "imageSetState": "LOCKED",
    "imageSetWorkflowStatus": "UPDATING",
    "createdAt": 1680027126.436,
    "updatedAt": 1680042257.908
}
```

Exemple 5 : pour ajouter un élément de données DICOM privé à une instance

L'`update-image-set-metadata`exemple suivant montre comment ajouter un élément privé à une instance spécifiée dans un ensemble d'images. La norme DICOM autorise les éléments de données privés pour la communication d'informations qui ne peuvent pas être contenus dans des

éléments de données standard. Vous pouvez créer, mettre à jour et supprimer des éléments de données privés à l'aide de UpdateImageSetMetadata cette action.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --force \
    --update-image-set-metadata-updates file://metadata-updates.json
```

Contenu de `metadata-updates.json`

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\":
 {\"001910F9\": \"97\"},\"DICOMVRs\": {\"001910F9\": \"DS\"}}}}}}"
    }
}
```

Sortie :

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 6 : Pour mettre à jour un élément de données DICOM privé vers une instance

L'`update-image-set-metadata`exemple suivant montre comment mettre à jour la valeur d'un élément de données privé appartenant à une instance au sein d'un ensemble d'images.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
```

```
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --force \
    --update-image-set-metadata-updates file://metadata-updates.json
```

Contenu de `metadata-updates.json`

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\":
 {\"00091001\": \"GE_GENESIS_DD\"}}}}}}"
    }
}
```

Sortie :

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

Exemple 7 : Pour mettre à jour un SOPInstance UID avec le paramètre force

L'`update-image-set-metadata`exemple suivant montre comment mettre à jour un SOPInstance UID en utilisant le paramètre force pour contourner les contraintes de métadonnées DICOM.

```
aws medical-imaging update-image-set-metadata \
        --datastore-id 12345678901234567890123456789012 \
        --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
        --latest-version-id 1 \
        --cli-binary-format raw-in-base64-out \
        --force \
```

```
        --update-image-set-metadata-updates file://metadata-updates.json
```

Contenu de `metadata-updates.json`

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"Series\":
{\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3656.0\":{\"Instances
\":{\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.0\":{\"DICOM\":
{\"SOPInstanceUID\":
\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.9\"}}}}}}"
    }
}
```

Sortie :

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

Pour plus d'informations, consultez la section Mise à jour des métadonnées des ensembles d'images dans le manuel du AWS HealthImaging développeur.

- Pour plus de détails sur l'API, reportez-vous UpdateImageSetMetadataà la section Référence des AWS CLI commandes.

HealthLake exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with HealthLake.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-fhir-datastore`

L'exemple de code suivant montre comment utiliser`create-fhir-datastore`.

AWS CLI

Exemple 1 : créer un magasin de données compatible SIGv4 HealthLake

L'`create-fhir-datastore`exemple suivant montre comment créer un nouveau magasin de données dans AWS HealthLake.

```
aws healthlake create-fhir-datastore \
    --datastore-type-version R4 \
    --datastore-name "FhirTestDatastore"
```

Sortie :

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "CREATING",
    "DatastoreId": "(Data store ID)"
}
```

Exemple 2 : créer un magasin de données compatible SMART sur FHIR HealthLake

L'`create-fhir-datastore`exemple suivant montre comment créer un nouveau magasin de données compatible SMART on FHIR dans. AWS HealthLake

```
aws healthlake create-fhir-datastore \
```

```
    --datastore-name "your-data-store-name" \
    --datastore-type-version R4 \
    --preload-data-config PreloadDataType="SYNTHEA" \
    --sse-configuration '{ "KmsEncryptionConfig": {  "CmkType":
 "CUSTOMER_MANAGED_KMS_KEY", "KmsKeyId": "arn:aws:kms:us-east-1:your-account-id:key/
your-key-id" } }' \
    --identity-provider-configuration  file://identity_provider_configuration.json
```

Contenu de `identity_provider_configuration.json` :

```
{
    "AuthorizationStrategy": "SMART_ON_FHIR_V1",
    "FineGrainedAuthorizationEnabled": true,
    "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-
lambda-name",
    "Metadata": "{\"issuer\":\"https://ehr.example.com\", \"jwks_uri\":\"https://
ehr.example.com/.well-known/jwks.json\",\"authorization_endpoint\":\"https://
ehr.example.com/auth/authorize\",\"token_endpoint\":\"https://ehr.token.com/auth/
token\",\"token_endpoint_auth_methods_supported\":[\"client_secret_basic\",\"foo\"],
\"grant_types_supported\":[\"client_credential\",\"foo\"],\"registration_endpoint\":
\"https://ehr.example.com/auth/register\",\"scopes_supported\":[\"openId\",\"profile
\",\"launch\"],\"response_types_supported\":[\"code\"],\"management_endpoint\":
\"https://ehr.example.com/user/manage\",\"introspection_endpoint\":\"https://
ehr.example.com/user/introspect\",\"revocation_endpoint\":\"https://ehr.example.com/
user/revoke\",\"code_challenge_methods_supported\":[\"S256\"],\"capabilities\":
[\"launch-ehr\",\"sso-openid-connect\",\"client-public\"]}"
}
```

Sortie :

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "CREATING",
    "DatastoreId": "(Data store ID)"
}
```

Pour plus d'informations, voir Création et surveillance d'un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, voir Create FHIRDatastore in AWS CLI Command Reference.

`delete-fhir-datastore`

L'exemple de code suivant montre comment utiliser`delete-fhir-datastore`.

AWS CLI

Pour supprimer un magasin de données FHIR

L'`delete-fhir-datastore`exemple suivant montre comment supprimer un magasin de données et tout son contenu dans AWS HealthLake.

```
aws healthlake delete-fhir-datastore \
    --datastore-id (Data store ID)
```

Sortie :

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "DELETING",
    "DatastoreId": "(Data store ID)"
}
```

Pour plus d'informations, voir Création et surveillance d'un magasin de données FHIR < https:// docs.aws.amazon.com/healthlake/ latest/devguide/working -with-fhir-healthlake.html> dans le guide du développeur.AWS HealthLake

- Pour plus de détails sur l'API, voir Supprimer FHIRDatastore dans le manuel de référence des AWS CLI commandes.

`describe-fhir-datastore`

L'exemple de code suivant montre comment utiliser`describe-fhir-datastore`.

AWS CLI

Pour décrire un magasin de données FHIR

L'`describe-fhir-datastore`exemple suivant montre comment rechercher les propriétés d'un magasin de données dans AWS HealthLake.

```
aws healthlake describe-fhir-datastore \
    --datastore-id "1f2f459836ac6c513ce899f9e4f66a59"
```

Sortie :

```
{
    "DatastoreProperties": {
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        },
        "SseConfiguration": {
            "KmsEncryptionConfig": {
                "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
                "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }
        },
        "DatastoreName": "Demo",
        "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Data store ID>",
        "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Data store ID>/r4/",
        "DatastoreStatus": "ACTIVE",
        "DatastoreTypeVersion": "R4",
        "CreatedAt": 1603761064.881,
        "DatastoreId": "<Data store ID>",
        "IdentityProviderConfiguration": {
            "AuthorizationStrategy": "AWS_AUTH",
            "FineGrainedAuthorizationEnabled": false
        }
    }
}
```

Pour plus d'informations, voir Création et surveillance d'un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, voir Description FHIRDatastore dans AWS CLI la référence des commandes.

`describe-fhir-export-job`

L'exemple de code suivant montre comment utiliser`describe-fhir-export-job`.

AWS CLI

Pour décrire une tâche d'exportation FHIR

L'`describe-fhir-export-job`exemple suivant montre comment rechercher les propriétés d'une tâche d'exportation FHIR dans AWS HealthLake.

```
aws healthlake describe-fhir-export-job \
    --datastore-id (Data store ID) \
    --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

Sortie :

```
{
    "ExportJobProperties": {
        "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
        "JobStatus": "IN_PROGRESS",
        "JobId": "9009813e9d69ba7cf79bcb3468780f16",
        "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
        "EndTime": "2024-11-20T11:34:01.636000-05:00",
        "OutputDataConfig": {
            "S3Configuration": {
            "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
            "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-
a250-f4c43ef46e83"
        }

        },
        "DatastoreId": "(Data store ID)"
    }
}
```

Pour plus d'informations, consultez la section Exportation de fichiers depuis un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, voir FHIRExportDescribe Job dans AWS CLI Command Reference.

`describe-fhir-import-job`

L'exemple de code suivant montre comment utiliser`describe-fhir-import-job`.

AWS CLI

Pour décrire une tâche d'importation FHIR

L'`describe-fhir-import-job`exemple suivant montre comment apprendre les propriétés d'une tâche d'importation FHIR à l'aide AWS HealthLake de.

```
aws healthlake describe-fhir-import-job \
    --datastore-id (Data store ID) \
    --job-id c145fbb27b192af392f8ce6e7838e34f
```

Sortie :

```
{
    "ImportJobProperties": {
    "InputDataConfig": {
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
        { "arrayitem2": 2 }
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "COMPLETED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",
    "SubmitTime": 1606272542.161,
    "EndTime": 1606272609.497,
    "DatastoreId": "(Data store ID)"
    }
}
```

Pour plus d'informations, consultez la section Importation de fichiers dans un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, voir FHIRImportDescribe Job dans AWS CLI Command Reference.

`list-fhir-datastores`

L'exemple de code suivant montre comment utiliser`list-fhir-datastores`.

AWS CLI

Pour répertorier les magasins de données FHIR

L'`list-fhir-datastores`exemple suivant montre comment utiliser la commande et comment les utilisateurs peuvent filtrer les résultats en fonction de l'état du magasin de données dans AWS HealthLake.

```
aws healthlake list-fhir-datastores \
    --filter DatastoreStatus=ACTIVE
```

Sortie :

```
{
    "DatastorePropertiesList": [
    {
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        },
        "SseConfiguration": {
            "KmsEncryptionConfig": {
                "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
                "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }
        },
        "DatastoreName": "Demo",
        "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Data store ID>",
        "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Data store ID>/r4/",
        "DatastoreStatus": "ACTIVE",
        "DatastoreTypeVersion": "R4",
        "CreatedAt": 1603761064.881,
        "DatastoreId": "<Data store ID>",
        "IdentityProviderConfiguration": {
            "AuthorizationStrategy": "AWS_AUTH",
            "FineGrainedAuthorizationEnabled": false
        }
    }
    ]
}
```

Pour plus d'informations, voir Création et surveillance d'un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, consultez la section Liste FHIRDatastores dans AWS CLI la référence des commandes.

`list-fhir-export-jobs`

L'exemple de code suivant montre comment utiliser`list-fhir-export-jobs`.

AWS CLI

Pour répertorier tous les emplois d'exportation FHIR

L'`list-fhir-export-jobs`exemple suivant montre comment utiliser la commande pour afficher la liste des tâches d'exportation associées à un compte.

```
aws healthlake list-fhir-export-jobs \
    --datastore-id (Data store ID) \
    --submitted-before (DATE like 2024-10-13T19:00:00Z)\
    --submitted-after (DATE like 2020-10-13T19:00:00Z )\
    --job-name "FHIR-EXPORT" \
    --job-status SUBMITTED  \
    --max-results (Integer between 1 and 500)
```

Sortie :

```
{
    "ExportJobPropertiesList": [
        {
            "ExportJobProperties": {
                "OutputDataConfig": {
                    "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                    "S3Configuration": {
                        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                        "KmsKeyId": "(KmsKey Id)"
                    }
                },
                "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role
 Name)",
                "JobStatus": "COMPLETED",
                "JobId": "c145fbb27b192af392f8ce6e7838e34f",
                "JobName": "FHIR-EXPORT",
                "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
                "EndTime": "2024-11-20T11:34:01.636000-05:00",
```

```
                "DatastoreId": "(Data store ID)"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Exportation de fichiers depuis un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, consultez la section FHIRExportRépertorier les tâches dans AWS CLI Command Reference.

`list-fhir-import-jobs`

L'exemple de code suivant montre comment utiliser`list-fhir-import-jobs`.

AWS CLI

Pour répertorier toutes les tâches d'importation FHIR

L'`list-fhir-import-jobs`exemple suivant montre comment utiliser la commande pour afficher la liste de toutes les tâches d'importation associées à un compte.

```
aws healthlake list-fhir-import-jobs \
    --datastore-id (Data store ID) \
    --submitted-before (DATE like 2024-10-13T19:00:00Z) \
    --submitted-after (DATE like 2020-10-13T19:00:00Z ) \
    --job-name "FHIR-IMPORT" \
    --job-status SUBMITTED  \
    -max-results (Integer between 1 and 500)
```

Sortie :

```
{
    "ImportJobPropertiesList": [
        {
            "JobId": "c0fddbf76f238297632d4aebdbfc9ddf",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2024-11-20T10:08:46.813000-05:00",
            "EndTime": "2024-11-20T10:10:09.093000-05:00",
            "DatastoreId": "(Data store ID)",
```

```
            "InputDataConfig": {
                "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
            },
            "JobOutputDataConfig": {
                "S3Configuration": {
                    "S3Uri": "s3://(Bucket Name)/
import/6407b9ae4c2def3cb6f1a46a0c599ec0-FHIR_IMPORT-
c0fddbf76f238297632d4aebdbfc9ddf/",
                    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/b7f645cb-
e564-4981-8672-9e012d1ff1a0"
                }
            },
            "JobProgressReport": {
                "TotalNumberOfScannedFiles": 1,
                "TotalSizeOfScannedFilesInMB": 0.001798,
                "TotalNumberOfImportedFiles": 1,
                "TotalNumberOfResourcesScanned": 1,
                "TotalNumberOfResourcesImported": 1,
                "TotalNumberOfResourcesWithCustomerError": 0,
                "TotalNumberOfFilesReadWithCustomerError": 0,
                "Throughput": 0.0
            },
            "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)"
        }
    ]
}
```

Pour plus d'informations, consultez la section Importation de fichiers dans le magasin de données FHIR dans le guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, consultez la section FHIRImportRépertorier les tâches dans AWS CLI Command Reference.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un magasin de données

L'`list-tags-for-resource`exemple suivant répertorie les balises associées au magasin de données spécifié. :

```
aws healthlake list-tags-for-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe"
```

Sortie :

```
{
    "tags": {
        "key": "value",
        "key1": "value1"
    }
}
```

Pour plus d'informations, consultez la section consacrée au balisage des ressources AWS HealthLake dans le guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`start-fhir-export-job`

L'exemple de code suivant montre comment utiliser`start-fhir-export-job`.

AWS CLI

Pour démarrer une tâche d'exportation FHIR

L'`start-fhir-export-job`exemple suivant montre comment démarrer une tâche d'exportation FHIR en utilisant AWS HealthLake.

```
aws healthlake start-fhir-export-job \
    --output-data-config '{"S3Configuration": {"S3Uri":"s3://(Bucket Name)/(Prefix
 Name)/","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-
f4c43ef46e83"}}' \
    --datastore-id (Data store ID) \
    --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

Sortie :

```
{
```

```
    "DatastoreId": "(Data store ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "9b9a51943afaedd0a8c0c26c49135a31"
}
```

Pour plus d'informations, consultez la section Exportation de fichiers depuis un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, consultez Start FHIRExport Job dans AWS CLI Command Reference.

`start-fhir-import-job`

L'exemple de code suivant montre comment utiliser`start-fhir-import-job`.

AWS CLI

Pour démarrer une tâche d'importation FHIR

L'`start-fhir-import-job`exemple suivant montre comment démarrer une tâche d'importation FHIR en utilisant AWS HealthLake.

```
aws healthlake start-fhir-import-job \
    --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \
    --job-output-data-config '{"S3Configuration": {"S3Uri":"s3://(Bucket Name)/
(Prefix Name)/","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-
b56c-4216-a250-f4c43ef46e83"}}' \
    --datastore-id (Data store ID) \
    --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)"
```

Sortie :

```
{
    "DatastoreId": "(Data store ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

Pour plus d'informations, consultez la section Importation de fichiers dans un magasin de données FHIR dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, consultez Start FHIRImport Job dans AWS CLI Command Reference.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une balise au magasin de données

L'`tag-resource`exemple suivant montre comment ajouter une balise à un magasin de données.

```
aws healthlake tag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \
    --tags '[{"Key": "key1", "Value": "value1"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout d'une balise à un magasin de données dans le Guide du AWS HealthLake développeur. .

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'un magasin de données.

L'`untag-resource`exemple suivant montre comment supprimer des balises d'un magasin de données.

```
aws healthlake untag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
b91723d65c6fdeb1d26543a49d2ed1fa" \
    --tag-keys '["key1"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression de balises d'un magasin de données dans le Guide du AWS HealthLake développeur.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

HealthOmics exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with HealthOmics.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`abort-multipart-read-set-upload`

L'exemple de code suivant montre comment utiliser`abort-multipart-read-set-upload`.

AWS CLI

Pour arrêter le téléchargement d'un jeu de lecture en plusieurs parties

L'`abort-multipart-read-set-upload`exemple suivant arrête le téléchargement d'un jeu de lecture en plusieurs parties dans votre magasin de HealthOmics séquences.

```
aws omics abort-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, reportez-vous à la section Téléchargement direct vers un magasin de séquences dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir AbortMultipartReadSetUploadla section Référence des AWS CLI commandes.

`accept-share`

L'exemple de code suivant montre comment utiliser`accept-share`.

AWS CLI

Pour accepter un partage des données de la boutique d'analyse

L'`accept-share`exemple suivant accepte un partage des données du magasin HealthOmics d'analyse.

```
aws omics accept-share \
    ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Sortie :

```
{
    "status": "ACTIVATING"
}
```

Pour plus d'informations, consultez la section Partage entre comptes dans le guide de l'AWS HealthOmics utilisateur.

- Pour plus de détails sur l'API, voir AcceptSharela section Référence des AWS CLI commandes.

`batch-delete-read-set`

L'exemple de code suivant montre comment utiliser`batch-delete-read-set`.

AWS CLI

Pour supprimer plusieurs ensembles de lecture

L'`batch-delete-read-set`exemple suivant supprime deux ensembles de lecture.

```
aws omics batch-delete-read-set \
```

```
    --sequence-store-id 1234567890 \
    --ids 1234567890 0123456789
```

En cas d'erreur lors de la suppression de l'un des ensembles de lecture spécifiés, le service renvoie une liste d'erreurs.

```
{
    "errors": [
        {
            "code": "",
            "id": "0123456789",
            "message": "The specified readset does not exist."
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir BatchDeleteReadSetla section Référence des AWS CLI commandes.

`cancel-annotation-import-job`

L'exemple de code suivant montre comment utiliser`cancel-annotation-import-job`.

AWS CLI

Pour annuler une tâche d'importation d'annotations

L'`cancel-annotation-import-job`exemple suivant annule une tâche d'importation d'annotations avec ID. `04f57618-xmpl-4fd0-9349-e5a85aefb997`

```
aws omics cancel-annotation-import-job \
    --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir CancelAnnotationImportJobla section Référence des AWS CLI commandes.

`cancel-run`

L'exemple de code suivant montre comment utiliser`cancel-run`.

AWS CLI

Pour annuler une course

L'`cancel-run`exemple suivant annule une exécution avec ID. `1234567`

```
aws omics cancel-run \
    --id 1234567
```

Pour plus d'informations, voir Exécuter le cycle de vie dans un flux de travail dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir CancelRunla section Référence des AWS CLI commandes.

`cancel-variant-import-job`

L'exemple de code suivant montre comment utiliser`cancel-variant-import-job`.

AWS CLI

Pour annuler une tâche d'importation de variantes

L'`cancel-variant-import-job`exemple suivant annule une tâche d'importation de variantes avec ID. `69cb65d6-xmpl-4a4a-9025-4565794b684e`

```
aws omics cancel-variant-import-job \
    --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir CancelVariantImportJobla section Référence des AWS CLI commandes.

`complete-multipart-read-set-upload`

L'exemple de code suivant montre comment utiliser`complete-multipart-read-set-upload`.

AWS CLI

Pour terminer un téléchargement en plusieurs parties une fois que vous avez chargé tous les composants.

L'`complete-multipart-read-set-upload`exemple suivant conclut un téléchargement en plusieurs parties dans un magasin de séquences une fois que tous les composants ont été chargés.

```
aws omics complete-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/
Vobh5zBDb4W4=","partNumber":1,"partSource":"SOURCE1"}]'
```

Sortie :

```
{
    "readSetId": "0000000001"
    "readSetId": "0000000002"
    "readSetId": "0000000003"
}
```

Pour plus d'informations, reportez-vous à la section Téléchargement direct vers un magasin de séquences dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir CompleteMultipartReadSetUploadla section Référence des AWS CLI commandes.

`create-annotation-store-version`

L'exemple de code suivant montre comment utiliser`create-annotation-store-version`.

AWS CLI

Pour créer une nouvelle version d'un magasin d'annotations

L'`create-annotation-store-version`exemple suivant crée une nouvelle version d'un magasin d'annotations.

```
aws omics create-annotation-store-version \
```

```
    --name my_annotation_store \
    --version-name my_version
```

Sortie :

```
{
    "creationTime": "2023-07-21T17:15:49.251040+00:00",
    "id": "3b93cdef69d2",
    "name": "my_annotation_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
    },
    "status": "CREATING",
    "versionName": "my_version"
}
```

Pour plus d'informations, consultez la section Création de nouvelles versions de magasins d'annotations dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir CreateAnnotationStoreVersionla section Référence des AWS CLI commandes.

`create-annotation-store`

L'exemple de code suivant montre comment utiliser`create-annotation-store`.

AWS CLI

Exemple 1 : pour créer un magasin d'annotations VCF

L'`create-annotation-store`exemple suivant crée un magasin d'annotations au format VCF.

```
aws omics create-annotation-store \
    --name my_ann_store \
    --store-format VCF \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

Sortie :

```
{
```

```
    "creationTime": "2022-11-23T22:48:39.226492Z",
    "id": "0a91xmplc71f",
    "name": "my_ann_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeFormat": "VCF"
}
```

Exemple 2 : pour créer un magasin d'annotations TSV

L'`create-annotation-store`exemple suivant crée un magasin d'annotations au format TSV.

```
aws omics create-annotation-store \
    --name tsv_ann_store \
    --store-format TSV \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890 \
    --store-options file://tsv-store-options.json
```

`tsv-store-options.json`configure les options de format pour les annotations.

```
{
    "tsvStoreOptions": {
        "annotationType": "CHR_START_END_ZERO_BASE",
        "formatToHeader": {
            "CHR": "chromosome",
            "START": "start",
            "END": "end"
        },
        "schema": [
            {
                "chromosome": "STRING"
            },
            {
                "start": "LONG"
            },
            {
                "end": "LONG"
            },
            {
```

```
                "name": "STRING"
            }
        ]
    }
}
```

Sortie :

```
{
    "creationTime": "2022-11-30T01:28:08.525586Z",
    "id": "861cxmpl96b0",
    "name": "tsv_ann_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeFormat": "TSV",
    "storeOptions": {
        "tsvStoreOptions": {
            "annotationType": "CHR_START_END_ZERO_BASE",
            "formatToHeader": {
                "CHR": "chromosome",
                "END": "end",
                "START": "start"
            },
            "schema": [
                {
                    "chromosome": "STRING"
                },
                {
                    "start": "LONG"
                },
                {
                    "end": "LONG"
                },
                {
                    "name": "STRING"
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir CreateAnnotationStorela section Référence des AWS CLI commandes.

`create-multipart-read-set-upload`

L'exemple de code suivant montre comment utiliser`create-multipart-read-set-upload`.

AWS CLI

Pour commencer le téléchargement d'un set de lecture en plusieurs parties.

L'`create-multipart-read-set-upload`exemple suivant lance un téléchargement d'un ensemble de lectures en plusieurs parties.

```
aws omics create-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --name HG00146 \
    --source-file-type FASTQ \
    --subject-id mySubject\
    --sample-id mySample\
    --description "FASTQ for HG00146"\
    --generated-from "1000 Genomes"
```

Sortie :

```
{
    "creationTime": "2022-07-13T23:25:20Z",
    "description": "FASTQ for HG00146",
    "generatedFrom": "1000 Genomes",
    "name": "HG00146",
    "sampleId": "mySample",
    "sequenceStoreId": "0123456789",
    "sourceFileType": "FASTQ",
    "subjectId": "mySubject",
    "uploadId": "1122334455"
}
```

Pour plus d'informations, reportez-vous à la section Téléchargement direct vers un magasin de séquences dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir CreateMultipartReadSetUploadla section Référence des AWS CLI commandes.

`create-reference-store`

L'exemple de code suivant montre comment utiliser`create-reference-store`.

AWS CLI

Pour créer un magasin de référence

L'`create-reference-store`exemple suivant crée un magasin de référence`my-ref-store`.

```
aws omics create-reference-store \
    --name my-ref-store
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
    "creationTime": "2022-11-22T22:13:25.947Z",
    "id": "1234567890",
    "name": "my-ref-store"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir CreateReferenceStorela section Référence des AWS CLI commandes.

`create-run-group`

L'exemple de code suivant montre comment utiliser`create-run-group`.

AWS CLI

Pour créer un groupe de course

L'`create-run-group`exemple suivant crée un groupe d'exécution nommé`cram-converter`.

```
aws omics create-run-group \
    --name cram-converter \
    --max-cpus 20 \
    --max-gpus 10 \
    --max-duration 600 \
    --max-runs 5
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "id": "1234567",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Création de groupes d'exécution dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir CreateRunGroupla section Référence des AWS CLI commandes.

`create-sequence-store`

L'exemple de code suivant montre comment utiliser`create-sequence-store`.

AWS CLI

Pour créer un magasin de séquences

L'`create-sequence-store`exemple suivant crée un magasin de séquences.

```
aws omics create-sequence-store \
    --name my-seq-store
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T01:24:33.629Z",
    "id": "1234567890",
    "name": "my-seq-store"
```

```
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir CreateSequenceStorela section Référence des AWS CLI commandes.

`create-share`

L'exemple de code suivant montre comment utiliser`create-share`.

AWS CLI

Pour créer un partage d'une boutique HealthOmics d'analyses

L'`create-share`exemple suivant montre comment créer un partage d'un magasin HealthOmics d'analyses qui peut être accepté par un abonné extérieur au compte.

```
aws omics create-share \
    --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
    --principal-subscriber "123456789012" \
    --name "my_Share-123"
```

Sortie :

```
{
    "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
    "name": "my_Share-123",
    "status": "PENDING"
}
```

Pour plus d'informations, consultez la section Partage entre comptes dans le guide de l'AWS HealthOmics utilisateur.

- Pour plus de détails sur l'API, voir CreateSharela section Référence des AWS CLI commandes.

`create-variant-store`

L'exemple de code suivant montre comment utiliser`create-variant-store`.

AWS CLI

Pour créer un magasin de variantes

L'`create-variant-store`exemple suivant crée un magasin de variantes nommé`my_var_store`.

```
aws omics create-variant-store \
    --name my_var_store \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

Sortie :

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, voir CreateVariantStorela section Référence des AWS CLI commandes.

`create-workflow`

L'exemple de code suivant montre comment utiliser`create-workflow`.

AWS CLI

Pour créer un flux de travail

L'`create-workflow`exemple suivant crée un flux de travail WDL.

```
aws omics create-workflow \
```

```
    --name cram-converter \
    --engine WDL \
    --definition-zip fileb://workflow-crambam.zip \
    --parameter-template file://workflow-params.json
```

`workflow-crambam.zip`est une archive ZIP contenant une définition de flux de travail. `workflow-params.json`définit les paramètres d'exécution du flux de travail.

```
{
    "ref_fasta" : {
        "description": "Reference genome fasta file",
        "optional": false
    },
    "ref_fasta_index" : {
        "description": "Index of the reference genome fasta file",
        "optional": false
    },
    "ref_dict" : {
        "description": "dictionary file for 'ref_fasta'",
        "optional": false
    },
    "input_cram" : {
        "description": "The Cram file to convert to BAM",
        "optional": false
    },
    "sample_name" : {
        "description": "The name of the input sample, used to name the output BAM",
        "optional": false
    }
}
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
    "id": "1234567",
    "status": "CREATING",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Création de flux de travail privés dans le guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir CreateWorkflowla section Référence des AWS CLI commandes.

`delete-annotation-store-versions`

L'exemple de code suivant montre comment utiliser`delete-annotation-store-versions`.

AWS CLI

Pour supprimer une version du magasin d'annotations

L'`delete-annotation-store-versions`exemple suivant supprime une version du magasin d'annotations.

```
aws omics delete-annotation-store-versions \
    --name my_annotation_store \
    --versions my_version
```

Sortie :

```
{
    "errors": []
}
```

Pour plus d'informations, consultez la section Création de nouvelles versions de magasins d'annotations dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, voir DeleteAnnotationStoreVersionsla section Référence des AWS CLI commandes.

`delete-annotation-store`

L'exemple de code suivant montre comment utiliser`delete-annotation-store`.

AWS CLI

Pour supprimer un magasin d'annotations

L'`delete-annotation-store`exemple suivant supprime un magasin d'annotations nommé`my_vcf_store`.

```
aws omics delete-annotation-store \
    --name my_vcf_store
```

Sortie :

```
{
    "status": "DELETING"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteAnnotationStoreà la section Référence des AWS CLI commandes.

`delete-reference-store`

L'exemple de code suivant montre comment utiliser`delete-reference-store`.

AWS CLI

Pour supprimer un magasin de référence

L'`delete-reference-store`exemple suivant supprime un magasin de référence avec un ID. `1234567890`

```
aws omics delete-reference-store \
    --id 1234567890
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteReferenceStoreà la section Référence des AWS CLI commandes.

`delete-reference`

L'exemple de code suivant montre comment utiliser`delete-reference`.

AWS CLI

Pour supprimer une référence

L'`delete-reference`exemple suivant supprime une référence.

```
aws omics delete-reference \
    --reference-store-id 1234567890 \
    --id 1234567890
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteReferenceà la section Référence des AWS CLI commandes.

`delete-run-group`

L'exemple de code suivant montre comment utiliser`delete-run-group`.

AWS CLI

Pour supprimer un groupe de course

L'`delete-run-group`exemple suivant supprime un groupe d'exécution avec un ID. `1234567`

```
aws omics delete-run-group \
    --id 1234567
```

Pour plus d'informations, consultez la section Suppression de courses et de groupes d'exécutions dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteRunGroupà la section Référence des AWS CLI commandes.

`delete-run`

L'exemple de code suivant montre comment utiliser`delete-run`.

AWS CLI

Pour supprimer une exécution de flux de travail

L'`delete-run`exemple suivant supprime une exécution avec un ID. `1234567`

```
aws omics delete-run \
    --id 1234567
```

Pour plus d'informations, consultez la section Suppression de courses et de groupes d'exécutions dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteRunà la section Référence des AWS CLI commandes.

`delete-sequence-store`

L'exemple de code suivant montre comment utiliser`delete-sequence-store`.

AWS CLI

Pour supprimer un magasin de séquences

L'`delete-sequence-store`exemple suivant supprime un magasin de séquences avec ID. `1234567890`

```
aws omics delete-sequence-store \
    --id 1234567890
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteSequenceStoreà la section Référence des AWS CLI commandes.

`delete-share`

L'exemple de code suivant montre comment utiliser`delete-share`.

AWS CLI

Pour supprimer un partage de données HealthOmics analytiques

L'`delete-share`exemple suivant supprime un partage entre comptes de données d'analyse.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Sortie :

```
{
    "status": "DELETING"
}
```

Pour plus d'informations, consultez la section Partage entre comptes dans le guide de l'AWS HealthOmics utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteShareà la section Référence des AWS CLI commandes.

`delete-variant-store`

L'exemple de code suivant montre comment utiliser`delete-variant-store`.

AWS CLI

Pour supprimer un magasin de variantes

L'`delete-variant-store`exemple suivant supprime un magasin de variantes nommé`my_var_store`.

```
aws omics delete-variant-store \
    --name my_var_store
```

Sortie :

```
{
    "status": "DELETING"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteVariantStoreà la section Référence des AWS CLI commandes.

`delete-workflow`

L'exemple de code suivant montre comment utiliser`delete-workflow`.

AWS CLI

Pour supprimer un flux de travail

L'`delete-workflow`exemple suivant supprime un flux de travail avec un ID. `1234567`

```
aws omics delete-workflow \
    --id 1234567
```

Pour plus d'informations, voir Supprimer un flux de travail privé dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteWorkflowà la section Référence des AWS CLI commandes.

`get-annotation-import-job`

L'exemple de code suivant montre comment utiliser`get-annotation-import-job`.

AWS CLI

Pour afficher une tâche d'importation d'annotations

L'`get-annotation-import-job`exemple suivant fournit des informations sur une tâche d'importation d'annotations.

```
aws omics get-annotation-import-job \
    --job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

Sortie :

```
{
    "creationTime": "2022-11-30T01:40:11.017746Z",
    "destinationName": "tsv_ann_store",
    "id": "984162c7-xmpl-4d23-ab47-286f7950bfbf",
    "items": [
        {
            "jobStatus": "COMPLETED",
            "source": "s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz"
```

```
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2022-11-30T01:42:39.134009Z"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetAnnotationImportJobà la section Référence des AWS CLI commandes.

`get-annotation-store-version`

L'exemple de code suivant montre comment utiliser`get-annotation-store-version`.

AWS CLI

Pour récupérer les métadonnées d'une version du magasin d'annotations

L'`get-annotation-store-version`exemple suivant extrait les métadonnées de la version du magasin d'annotations demandée.

```
aws omics get-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version
```

Sortie :

```
{
    "storeId": "4934045d1c6d",
    "id": "2a3f4a44aa7b",
    "status": "ACTIVE",
    "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version",
    "name": "my_annotation_store",
    "versionName": "my_version",
    "creationTime": "2023-07-21T17:15:49.251040+00:00",
    "updateTime": "2023-07-21T17:15:56.434223+00:00",
    "statusMessage": "",
```

```
    "versionSizeBytes": 0
}
```

Pour plus d'informations, consultez la section Création de nouvelles versions de magasins d'annotations dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetAnnotationStoreVersionà la section Référence des AWS CLI commandes.

`get-annotation-store`

L'exemple de code suivant montre comment utiliser`get-annotation-store`.

AWS CLI

Pour afficher un magasin d'annotations

L'`get-annotation-store`exemple suivant permet d'obtenir des informations sur un magasin d'annotations nommé`my_ann_store`.

```
aws omics get-annotation-store \
    --name my_ann_store
```

Sortie :

```
{
    "creationTime": "2022-11-23T22:48:39.226492Z",
    "id": "0a91xmplc71f",
    "name": "my_ann_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/my_ann_store",
    "storeFormat": "VCF",
    "storeSizeBytes": 0,
    "tags": {}
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetAnnotationStoreà la section Référence des AWS CLI commandes.

`get-read-set-activation-job`

L'exemple de code suivant montre comment utiliser`get-read-set-activation-job`.

AWS CLI

Pour consulter une tâche d'activation du Read Set

L'`get-read-set-activation-job`exemple suivant fournit des informations sur une tâche d'activation d'un ensemble de lecture.

```
aws omics get-read-set-activation-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

Sortie :

```
{
    "completionTime": "2022-12-06T22:33:42.828Z",
    "creationTime": "2022-12-06T22:32:45.213Z",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "sources": [
        {
            "readSetId": "1234567890",
            "status": "FINISHED",
            "statusMessage": "No activation needed as read set is already in
 ACTIVATING or ACTIVE state."
        }
    ],
    "status": "COMPLETED",
    "statusMessage": "The job completed successfully."
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReadSetActivationJobà la section Référence des AWS CLI commandes.

`get-read-set-export-job`

L'exemple de code suivant montre comment utiliser`get-read-set-export-job`.

AWS CLI

Pour afficher une tâche d'exportation d'ensembles de lectures

L'`get-read-set-export-job`exemple suivant fournit des informations sur une tâche d'exportation d'ensembles de lectures.

```
aws omics get-read-set-export-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

Sortie :

```
{
    "completionTime": "2022-12-06T22:39:14.491Z",
    "creationTime": "2022-12-06T22:37:18.612Z",
    "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "COMPLETED",
    "statusMessage": "The job is submitted and will start soon."
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReadSetExportJobà la section Référence des AWS CLI commandes.

`get-read-set-import-job`

L'exemple de code suivant montre comment utiliser`get-read-set-import-job`.

AWS CLI

Pour afficher une tâche d'importation d'ensembles de lectures

L'`get-read-set-import-job`exemple suivant fournit des informations sur une tâche d'importation d'ensembles de lectures.

```
aws omics get-read-set-import-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

Sortie :

```
{
    "creationTime": "2022-11-23T01:36:38.158Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "sources": [
        {
            "name": "HG00100",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "bam-sample",
            "sourceFileType": "BAM",
            "sourceFiles": {
                "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam",
                "source2": ""
            },
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress.",
            "subjectId": "bam-subject",
            "tags": {
                "aws:omics:sampleId": "bam-sample",
                "aws:omics:subjectId": "bam-subject"
            }
        },
        {
            "name": "HG00146",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "fastq-sample",
            "sourceFileType": "FASTQ",
            "sourceFiles": {
                "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_1.filt.fastq.gz",
                "source2": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_2.filt.fastq.gz"
```

```
            },
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress.",
            "subjectId": "fastq-subject",
            "tags": {
                "aws:omics:sampleId": "fastq-sample",
                "aws:omics:subjectId": "fastq-subject"
            }
        },
        {
            "name": "HG00096",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "cram-sample",
            "sourceFileType": "CRAM",
            "sourceFiles": {
                "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram",
                "source2": ""
            },
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress.",
            "subjectId": "cram-subject",
            "tags": {
                "aws:omics:sampleId": "cram-sample",
                "aws:omics:subjectId": "cram-subject"
            }
        }
    ],
    "status": "IN_PROGRESS",
    "statusMessage": "The job is currently in progress."
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReadSetImportJobà la section Référence des AWS CLI commandes.

`get-read-set-metadata`

L'exemple de code suivant montre comment utiliser`get-read-set-metadata`.

AWS CLI

Pour consulter un set de lecture

L'`get-read-set-metadata`exemple suivant permet d'obtenir des informations sur les fichiers d'un ensemble de lectures.

```
aws omics get-read-set-metadata \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/1234567890",
    "creationTime": "2022-11-23T21:55:00.515Z",
    "fileType": "FASTQ",
    "files": {
        "source1": {
            "contentLength": 310054739,
            "partSize": 104857600,
            "totalParts": 3
        },
        "source2": {
            "contentLength": 307846621,
            "partSize": 104857600,
            "totalParts": 3
        }
    },
    "id": "1234567890",
    "name": "HG00146",
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
    "sampleId": "fastq-sample",
    "sequenceInformation": {
        "alignment": "UNALIGNED",
        "totalBaseCount": 677717384,
        "totalReadCount": 8917334
    },
    "sequenceStoreId": "1234567890",
    "status": "ACTIVE",
    "subjectId": "fastq-subject"
```

```
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReadSetMetadataà la section Référence des AWS CLI commandes.

`get-read-set`

L'exemple de code suivant montre comment utiliser`get-read-set`.

AWS CLI

Pour télécharger un kit de lecture

L'`get-read-set`exemple suivant télécharge la partie 3 d'un ensemble de lecture en tant que`1234567890.3.bam`.

```
aws omics get-read-set \
    --sequence-store-id 1234567890 \
    --id 1234567890 \
    --part-number 3  1234567890.3.bam
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReadSetà la section Référence des AWS CLI commandes.

`get-reference-import-job`

L'exemple de code suivant montre comment utiliser`get-reference-import-job`.

AWS CLI

Pour consulter une tâche d'importation de référence

L'`get-reference-import-job`exemple suivant permet d'obtenir des informations sur une tâche d'importation de référence.

```
aws omics get-reference-import-job \
    --reference-store-id 1234567890 \
    --id 1234567890
```

Sortie :

```
{
    "creationTime": "2022-11-22T22:25:41.124Z",
    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "sources": [
        {
            "name": "assembly-38",
            "sourceFile": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress."
        }
    ],
    "status": "IN_PROGRESS",
    "statusMessage": "The job is currently in progress."
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReferenceImportJobà la section Référence des AWS CLI commandes.

`get-reference-metadata`

L'exemple de code suivant montre comment utiliser`get-reference-metadata`.

AWS CLI

Pour consulter une référence

L'`get-reference-metadata`exemple suivant permet d'obtenir des détails sur une référence.

```
aws omics get-reference-metadata \
```

```
    --reference-store-id 1234567890 \
    --id 1234567890
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
    "creationTime": "2022-11-22T22:27:09.033Z",
    "files": {
        "index": {
            "contentLength": 160928,
            "partSize": 104857600,
            "totalParts": 1
        },
        "source": {
            "contentLength": 3249912778,
            "partSize": 104857600,
            "totalParts": 31
        }
    },
    "id": "1234567890",
    "md5": "7ff134953dcca8c8997453bbb80b6b5e",
    "name": "assembly-38",
    "referenceStoreId": "1234567890",
    "status": "ACTIVE",
    "updateTime": "2022-11-22T22:27:09.033Z"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReferenceMetadataà la section Référence des AWS CLI commandes.

`get-reference-store`

L'exemple de code suivant montre comment utiliser`get-reference-store`.

AWS CLI

Pour consulter un magasin de référence

L'`get-reference-store`exemple suivant permet d'obtenir des informations sur un magasin de référence.

```
aws omics get-reference-store \
    --id 1234567890
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
    "creationTime": "2022-09-23T23:27:20.364Z",
    "id": "1234567890",
    "name": "my-rstore-0"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReferenceStoreà la section Référence des AWS CLI commandes.

`get-reference`

L'exemple de code suivant montre comment utiliser`get-reference`.

AWS CLI

Pour télécharger une référence génomique

L'`get-reference`exemple suivant télécharge la partie 1 d'un génome en tant que`hg38.1.fa`.

```
aws omics get-reference \
    --reference-store-id 1234567890 \
    --id 1234567890 \
    --part-number 1 hg38.1.fa
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetReferenceà la section Référence des AWS CLI commandes.

`get-run-group`

L'exemple de code suivant montre comment utiliser`get-run-group`.

AWS CLI

Pour afficher un groupe de courses

L'`get-run-group`exemple suivant permet d'obtenir des informations sur un groupe d'exécution.

```
aws omics get-run-group \
    --id 1234567
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "creationTime": "2022-12-01T00:58:42.915219Z",
    "id": "1234567",
    "maxCpus": 20,
    "maxDuration": 600,
    "name": "cram-convert",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Création de groupes d'exécution dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetRunGroupà la section Référence des AWS CLI commandes.

`get-run-task`

L'exemple de code suivant montre comment utiliser`get-run-task`.

AWS CLI

Pour afficher une tâche

L'`get-run-task`exemple suivant fournit des informations sur une tâche de flux de travail.

```
aws omics get-run-task \
```

```
    --id 1234567 \
    --task-id 1234567
```

Sortie :

```
{
    "cpus": 1,
    "creationTime": "2022-11-30T23:13:00.718651Z",
    "logStream": "arn:aws:logs:us-west-2:123456789012:log-group:/aws/omics/
WorkflowLog:log-stream:run/1234567/task/1234567",
    "memory": 15,
    "name": "CramToBamTask",
    "startTime": "2022-11-30T23:17:47.016Z",
    "status": "COMPLETED",
    "stopTime": "2022-11-30T23:18:21.503Z",
    "taskId": "1234567"
}
```

Pour plus d'informations, consultez la section Cycle de vie des tâches lors d'une HealthOmics exécution dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetRunTaskà la section Référence des AWS CLI commandes.

`get-run`

L'exemple de code suivant montre comment utiliser`get-run`.

AWS CLI

Pour afficher l'exécution d'un flux de travail

L'`get-run`exemple suivant fournit des informations sur l'exécution d'un flux de travail.

```
aws omics get-run \
    --id 1234567
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "creationTime": "2022-11-30T22:58:22.615865Z",
```

```
    "digest":
 "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
    "id": "1234567",
    "name": "cram-to-bam",
    "outputUri": "s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/",
    "parameters": {
        "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
        "ref_fasta_index": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta.fai",
        "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
        "sample_name": "NA12878",
        "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram"
    },
    "resourceDigests": {
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta.fai":
 "etag:f76371b113734a56cde236bc0372de0a",
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.dict":
 "etag:3884c62eb0e53fa92459ed9bff133ae6",
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta":
 "etag:e307d81c605fb91b7720a08f00276842-388",
        "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram":
 "etag:a9f52976381286c6143b5cc681671ec6"
    },
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "startedBy": "arn:aws:iam::123456789012:user/laptop-2020",
    "status": "STARTING",
    "tags": {},
    "workflowId": "1234567",
    "workflowType": "PRIVATE"
}
```

Pour plus d'informations, voir Exécuter le cycle de vie dans un flux de travail dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetRunà la section Référence des AWS CLI commandes.

`get-sequence-store`

L'exemple de code suivant montre comment utiliser`get-sequence-store`.

AWS CLI

Pour afficher un magasin de séquences

L'`get-sequence-store`exemple suivant permet d'obtenir des informations sur un magasin de séquences avec ID`1234567890`.

```
aws omics get-sequence-store \
    --id 1234567890
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T19:55:48.376Z",
    "id": "1234567890",
    "name": "my-seq-store"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetSequenceStoreà la section Référence des AWS CLI commandes.

`get-share`

L'exemple de code suivant montre comment utiliser`get-share`.

AWS CLI

Pour récupérer les métadonnées relatives à un partage de HealthOmics données d'analyse

L'`get-share`exemple suivant extrait les métadonnées d'un partage entre comptes de données d'analyse.

```
aws omics get-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Sortie :

```
{
    "share": {
        "shareId": "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a",
        "name": "my_Share-123",
        "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store",
        "principalSubscriber": "123456789012",
        "ownerId": "555555555555",
        "status": "PENDING"
    }
}
```

Pour plus d'informations, consultez la section Partage entre comptes dans le guide de l'AWS HealthOmics utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetShareà la section Référence des AWS CLI commandes.

`get-variant-import-job`

L'exemple de code suivant montre comment utiliser`get-variant-import-job`.

AWS CLI

Pour afficher une tâche d'importation de variantes

L'`get-variant-import-job`exemple suivant fournit des informations sur une tâche d'importation de variantes.

```
aws omics get-variant-import-job \
    --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

Sortie :

```
{
    "creationTime": "2022-11-23T22:42:50.037812Z",
    "destinationName": "my_var_store",
    "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",
    "items": [
        {
            "jobStatus": "IN_PROGRESS",
```

```
            "source": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.known_indels.vcf.gz"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "runLeftNormalization": false,
    "status": "IN_PROGRESS",
    "updateTime": "2022-11-23T22:43:05.898309Z"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetVariantImportJobà la section Référence des AWS CLI commandes.

`get-variant-store`

L'exemple de code suivant montre comment utiliser`get-variant-store`.

AWS CLI

Pour consulter un magasin de variantes

L'`get-variant-store`exemple suivant permet d'obtenir des informations sur un magasin de variantes.

```
aws omics get-variant-store \
    --name my_var_store
```

Sortie :

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
```

```
    "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/my_var_store",
    "storeSizeBytes": 0,
    "tags": {},
    "updateTime": "2022-11-23T22:09:24.931711Z"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetVariantStoreà la section Référence des AWS CLI commandes.

`get-workflow`

L'exemple de code suivant montre comment utiliser`get-workflow`.

AWS CLI

Pour consulter un flux de travail

L'`get-workflow`exemple suivant fournit des détails sur un flux de travail avec ID`1234567`.

```
aws omics get-workflow \
    --id 1234567
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
    "creationTime": "2022-11-30T22:33:16.225368Z",
    "digest":
 "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
    "engine": "WDL",
    "id": "1234567",
    "main": "workflow-crambam.wdl",
    "name": "cram-converter",
    "parameterTemplate": {
        "ref_dict": {
            "description": "dictionary file for 'ref_fasta'"
        },
        "ref_fasta_index": {
            "description": "Index of the reference genome fasta file"
```

```
        },
        "ref_fasta": {
            "description": "Reference genome fasta file"
        },
        "input_cram": {
            "description": "The Cram file to convert to BAM"
        },
        "sample_name": {
            "description": "The name of the input sample, used to name the output
 BAM"
        }
    },
    "status": "ACTIVE",
    "statusMessage": "workflow-crambam.wdl\n    workflow CramToBamFlow\n
 call CramToBamTask\n        call ValidateSamFile\n    task CramToBamTask\n    task
 ValidateSamFile\n",
    "tags": {},
    "type": "PRIVATE"
}
```

Pour plus d'informations, consultez la section Création de flux de travail privés dans le guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetWorkflowà la section Référence des AWS CLI commandes.

`list-annotation-import-jobs`

L'exemple de code suivant montre comment utiliser`list-annotation-import-jobs`.

AWS CLI

Pour obtenir une liste des tâches d'importation d'annotations

Vous trouverez ci-dessous `list-annotation-import-jobs` une liste des tâches d'importation d'annotations.

```
aws omics list-annotation-import-jobs
```

Sortie :

```
{
```

```
    "annotationImportJobs": [
        {
            "creationTime": "2022-11-30T01:39:41.478294Z",
            "destinationName": "gff_ann_store",
            "id": "18a9e792-xmpl-4869-a105-e5b602900444",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-30T01:47:09.145178Z"
        },
        {
            "creationTime": "2022-11-30T00:45:58.007838Z",
            "destinationName": "my_ann_store",
            "id": "4e9eafc8-xmpl-431e-a0b2-3bda27cb600a",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "FAILED",
            "updateTime": "2022-11-30T00:47:01.706325Z"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListAnnotationImportJobsà la section Référence des AWS CLI commandes.

`list-annotation-store-versions`

L'exemple de code suivant montre comment utiliser`list-annotation-store-versions`.

AWS CLI

Pour répertorier toutes les versions d'un magasin d'annotations.

L'`list-annotation-store-versions`exemple suivant répertorie toutes les versions existantes d'un magasin d'annotations.

```
aws omics list-annotation-store-versions \
```

```
    --name my_annotation_store
```

Sortie :

```
{
    "annotationStoreVersions": [
        {
        "storeId": "4934045d1c6d",
        "id": "2a3f4a44aa7b",
        "status": "CREATING",
        "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_2",
        "name": "my_annotation_store",
        "versionName": "my_version_2",
        "creation Time": "2023-07-21T17:20:59.380043+00:00",
        "versionSizeBytes": 0
},
{
     "storeId": "4934045d1c6d",
     "id": "4934045d1c6d",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_1",
     "name": "my_annotation_store",
     "versionName": "my_version_1",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
     }

}
```

Pour plus d'informations, consultez la section Création de nouvelles versions de magasins d'annotations dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListAnnotationStoreVersionsà la section Référence des AWS CLI commandes.

`list-annotation-stores`

L'exemple de code suivant montre comment utiliser`list-annotation-stores`.

AWS CLI

Pour obtenir la liste des magasins d'annotations

L'`list-annotation-stores`exemple suivant permet d'obtenir une liste des magasins d'annotations.

```
aws omics list-annotation-stores
```

Sortie :

```
{
    "annotationStores": [
        {
            "creationTime": "2022-11-23T22:48:39.226492Z",
            "id": "0a91xmplc71f",
            "name": "my_ann_store",
            "reference": {
                "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
            },
            "status": "ACTIVE",
            "statusMessage": "",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/
my_ann_store",
            "storeFormat": "VCF",
            "storeSizeBytes": 0,
            "updateTime": "2022-11-23T22:53:27.372840Z"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListAnnotationStoresà la section Référence des AWS CLI commandes.

`list-multipart-read-set-uploads`

L'exemple de code suivant montre comment utiliser`list-multipart-read-set-uploads`.

AWS CLI

Pour répertorier tous les téléchargements de sets de lecture partitionnés et leur statut.

L'`list-multipart-read-set-uploads`exemple suivant répertorie tous les téléchargements de sets de lecture partitionnés et leur statut.

```
aws omics list-multipart-read-set-uploads \
    --sequence-store-id 0123456789
```

Sortie :

```
{
"uploads":
    [
        {
           "sequenceStoreId": "0123456789",
           "uploadId": "8749584421",
           "sourceFileType": "FASTQ",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "generatedFrom": "1000 Genomes",
            "name": "HG00146",
            "description": "FASTQ for HG00146",
            "creationTime": "2023-11-29T19:22:51.349298+00:00"
        },
        {
            "sequenceStoreId": "0123456789",
            "uploadId": "5290538638",
            "sourceFileType": "BAM",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "generatedFrom": "1000 Genomes",
            "referenceArn": "arn:aws:omics:us-
west-2:845448930428:referenceStore/8168613728/reference/2190697383",
            "name": "HG00146",
            "description": "BAM for HG00146",
            "creationTime": "2023-11-29T19:23:33.116516+00:00"
        },
        {
            "sequenceStoreId": "0123456789",
            "uploadId": "4174220862",
            "sourceFileType": "BAM",
```

```
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "generatedFrom": "1000 Genomes",
            "referenceArn": "arn:aws:omics:us-
west-2:845448930428:referenceStore/8168613728/reference/2190697383",
            "name": "HG00147",
            "description": "BAM for HG00147",
            "creationTime": "2023-11-29T19:23:47.007866+00:00"
        }
    ]
}
```

Pour plus d'informations, reportez-vous à la section Téléchargement direct vers un magasin de séquences dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListMultipartReadSetUploadsà la section Référence des AWS CLI commandes.

`list-read-set-activation-jobs`

L'exemple de code suivant montre comment utiliser`list-read-set-activation-jobs`.

AWS CLI

Pour obtenir une liste des tâches d'activation du set de lecture

L'`list-read-set-activation-jobs`exemple suivant permet d'obtenir une liste de tâches d'activation pour un magasin de séquences avec id`1234567890`.

```
aws omics list-read-set-activation-jobs \
    --sequence-store-id 1234567890
```

Sortie :

```
{
    "activationJobs": [
        {
            "completionTime": "2022-12-06T22:33:42.828Z",
            "creationTime": "2022-12-06T22:32:45.213Z",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED"
```

```
        },
        {
            "creationTime": "2022-12-06T22:35:10.100Z",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "IN_PROGRESS"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReadSetActivationJobsà la section Référence des AWS CLI commandes.

`list-read-set-export-jobs`

L'exemple de code suivant montre comment utiliser`list-read-set-export-jobs`.

AWS CLI

Pour obtenir une liste des tâches d'exportation définies par des ensembles de lecture

L'`list-read-set-export-jobs`exemple suivant permet d'obtenir une liste de tâches d'exportation pour un magasin de séquences avec id`1234567890`.

```
aws omics list-read-set-export-jobs \
    --sequence-store-id 1234567890
```

Sortie :

```
{
    "exportJobs": [
        {
            "completionTime": "2022-12-06T22:39:14.491Z",
            "creationTime": "2022-12-06T22:37:18.612Z",
            "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED"
        },
```

```
        {
            "creationTime": "2022-12-06T22:38:04.871Z",
            "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "IN_PROGRESS"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReadSetExportJobsà la section Référence des AWS CLI commandes.

`list-read-set-import-jobs`

L'exemple de code suivant montre comment utiliser`list-read-set-import-jobs`.

AWS CLI

Pour obtenir une liste des tâches d'importation de sets de lecture

L'`list-read-set-import-jobs`exemple suivant permet d'obtenir une liste de tâches d'importation pour un magasin de séquences avec id`1234567890`.

```
aws omics list-read-set-import-jobs \
    --sequence-store-id 1234567890
```

Sortie :

```
{
    "importJobs": [
        {
            "completionTime": "2022-11-29T18:17:49.244Z",
            "creationTime": "2022-11-29T17:32:47.700Z",
            "id": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED"
```

```
        },
        {
            "completionTime": "2022-11-23T22:01:34.090Z",
            "creationTime": "2022-11-23T21:52:43.289Z",
            "id": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED_WITH_FAILURES"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReadSetImportJobsà la section Référence des AWS CLI commandes.

`list-read-set-upload-parts`

L'exemple de code suivant montre comment utiliser`list-read-set-upload-parts`.

AWS CLI

Pour répertorier toutes les pièces d'un téléchargement en plusieurs parties demandé pour un magasin de séquences.

L'`list-read-set-upload-parts`exemple suivant répertorie toutes les parties d'un téléchargement partitionné demandé pour un magasin de séquences.

```
aws omics list-read-set-upload-parts \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --part-source SOURCE1
```

Sortie :

```
{
    "parts": [
        {
            "partNumber": 1,
```

```
            "partSize": 94371840,
            "file": "SOURCE1",
            "checksum":
 "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
            "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
        }
        {
            "partNumber": 2,
            "partSize": 10471840,
            "file": "SOURCE1",
            "checksum":
 "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
            "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
        }
      ]

}
```

Pour plus d'informations, reportez-vous à la section Téléchargement direct vers un magasin de séquences dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListReadSetUploadPartsà la section Référence des AWS CLI commandes.

`list-read-sets`

L'exemple de code suivant montre comment utiliser`list-read-sets`.

AWS CLI

Pour obtenir une liste des ensembles de lecture

L'`list-read-sets`exemple suivant obtient une liste d'ensembles de lecture pour un magasin de séquences avec id`1234567890`.

```
aws omics list-read-sets \
    --sequence-store-id 1234567890
```

Sortie :

```
{
    "readSets": [
```

```
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/1234567890",
            "creationTime": "2022-11-23T21:55:00.515Z",
            "fileType": "FASTQ",
            "id": "1234567890",
            "name": "HG00146",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "fastq-sample",
            "sequenceStoreId": "1234567890",
            "status": "ACTIVE",
            "subjectId": "fastq-subject"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReadSetsà la section Référence des AWS CLI commandes.

`list-reference-import-jobs`

L'exemple de code suivant montre comment utiliser`list-reference-import-jobs`.

AWS CLI

Pour obtenir une liste des tâches d'importation de référence

L'`list-reference-import-jobs`exemple suivant permet d'obtenir une liste de tâches d'importation de référence pour un magasin de référence avec un identifiant`1234567890`.

```
aws omics list-reference-import-jobs \
    --reference-store-id 1234567890
```

Sortie :

```
{
    "importJobs": [
        {
```

```
            "completionTime": "2022-11-23T19:54:58.204Z",
            "creationTime": "2022-11-23T19:53:20.729Z",
            "id": "1234567890",
            "referenceStoreId": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "status": "COMPLETED"
        },
        {
            "creationTime": "2022-11-23T20:34:03.250Z",
            "id": "1234567890",
            "referenceStoreId": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "status": "IN_PROGRESS"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReferenceImportJobsà la section Référence des AWS CLI commandes.

`list-reference-stores`

L'exemple de code suivant montre comment utiliser`list-reference-stores`.

AWS CLI

Pour obtenir la liste des magasins de référence

L'`list-reference-stores`exemple suivant permet d'obtenir une liste de magasins de référence.

```
aws omics list-reference-stores
```

Sortie :

```
{
    "referenceStores": [
        {
```

```
            "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
            "creationTime": "2022-11-22T22:13:25.947Z",
            "id": "1234567890",
            "name": "my-ref-store"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReferenceStoresà la section Référence des AWS CLI commandes.

`list-references`

L'exemple de code suivant montre comment utiliser`list-references`.

AWS CLI

Pour obtenir une liste de références

L'`list-references`exemple suivant permet d'obtenir une liste de références génomiques pour un magasin de référence avec id`1234567890`.

```
aws omics list-references \
    --reference-store-id 1234567890
```

Sortie :

```
{
    "references": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
            "creationTime": "2022-11-22T22:27:09.033Z",
            "id": "1234567890",
            "md5": "7ff134953dcca8c8997453bbb80b6b5e",
            "name": "assembly-38",
            "referenceStoreId": "1234567890",
            "status": "ACTIVE",
            "updateTime": "2022-11-22T22:27:09.033Z"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListReferencesà la section Référence des AWS CLI commandes.

`list-run-groups`

L'exemple de code suivant montre comment utiliser`list-run-groups`.

AWS CLI

Pour obtenir une liste des groupes de course

L'`list-run-groups`exemple suivant permet d'obtenir une liste de groupes d'exécution.

```
aws omics list-run-groups
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
            "creationTime": "2022-12-01T00:58:42.915219Z",
            "id": "1234567",
            "maxCpus": 20,
            "maxDuration": 600,
            "name": "cram-convert"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création de groupes d'exécution dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListRunGroupsà la section Référence des AWS CLI commandes.

`list-run-tasks`

L'exemple de code suivant montre comment utiliser`list-run-tasks`.

AWS CLI

Pour obtenir une liste de tâches

L'`list-run-tasks`exemple suivant permet d'obtenir une liste de tâches pour une exécution de flux de travail.

```
aws omics list-run-tasks \
    --id 1234567
```

Sortie :

```
{
    "items": [
        {
            "cpus": 1,
            "creationTime": "2022-11-30T23:13:00.718651Z",
            "memory": 15,
            "name": "CramToBamTask",
            "startTime": "2022-11-30T23:17:47.016Z",
            "status": "COMPLETED",
            "stopTime": "2022-11-30T23:18:21.503Z",
            "taskId": "1234567"
        },
        {
            "cpus": 1,
            "creationTime": "2022-11-30T23:18:32.315606Z",
            "memory": 4,
            "name": "ValidateSamFile",
            "startTime": "2022-11-30T23:23:40.165Z",
            "status": "COMPLETED",
            "stopTime": "2022-11-30T23:24:14.766Z",
            "taskId": "1234567"
        }
    ]
}
```

Pour plus d'informations, consultez la section Cycle de vie des tâches lors d'une HealthOmics exécution dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListRunTasksà la section Référence des AWS CLI commandes.

`list-runs`

L'exemple de code suivant montre comment utiliser`list-runs`.

AWS CLI

Pour obtenir une liste des exécutions de flux de travail

L'`list-runs`exemple suivant permet d'obtenir une liste des exécutions de flux de travail.

```
aws omics list-runs
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
            "creationTime": "2022-12-02T23:20:01.202074Z",
            "id": "1234567",
            "name": "cram-to-bam",
            "priority": 1,
            "startTime": "2022-12-02T23:29:18.115Z",
            "status": "COMPLETED",
            "stopTime": "2022-12-02T23:57:54.428812Z",
            "storageCapacity": 10,
            "workflowId": "1234567"
        },
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
            "creationTime": "2022-12-03T00:16:57.180066Z",
            "id": "1234567",
            "name": "cram-to-bam",
            "priority": 1,
            "startTime": "2022-12-03T00:26:50.233Z",
            "status": "FAILED",
            "stopTime": "2022-12-03T00:37:21.451340Z",
            "storageCapacity": 10,
            "workflowId": "1234567"
        },
```

```
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
            "creationTime": "2022-12-05T17:57:08.444817Z",
            "id": "1234567",
            "name": "cram-to-bam",
            "status": "STARTING",
            "workflowId": "1234567"
        }
    ]
}
```

Pour plus d'informations, voir Exécuter le cycle de vie dans un flux de travail dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListRunsà la section Référence des AWS CLI commandes.

`list-sequence-stores`

L'exemple de code suivant montre comment utiliser`list-sequence-stores`.

AWS CLI

Pour obtenir une liste des magasins de séquences

L'`list-sequence-stores`exemple suivant permet d'obtenir une liste de magasins de séquences.

```
aws omics list-sequence-stores
```

Sortie :

```
{
    "sequenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
            "creationTime": "2022-11-23T01:24:33.629Z",
            "id": "1234567890",
            "name": "my-seq-store"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListSequenceStoresà la section Référence des AWS CLI commandes.

`list-shares`

L'exemple de code suivant montre comment utiliser`list-shares`.

AWS CLI

Pour répertorier les partages disponibles d'une donnée HealthOmics d'analyse

L'`list-shares`exemple suivant répertorie tous les partages créés pour un propriétaire de ressource.

```
aws omics list-shares \
    --resource-owner SELF
```

Sortie :

```
{
    "shares": [
        {
            "shareId": "595c1cbd-a008-4eca-a887-954d30c91c6e",
            "name": "myShare",
            "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_1",
            "principalSubscriber": "123456789012",
            "ownerId": "555555555555",
            "status": "PENDING"
        }
        {
            "shareId": "39b65d0d-4368-4a19-9814-b0e31d73c10a",
            "name": "myShare3456",
            "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_2",
            "principalSubscriber": "123456789012",
            "ownerId": "555555555555",
            "status": "ACTIVE"
        },
```

```
        {
            "shareId": "203152f5-eef9-459d-a4e0-a691668d44ef",
            "name": "myShare4",
            "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_3",
            "principalSubscriber": "123456789012",
            "ownerId": "555555555555",
            "status": "ACTIVE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Partage entre comptes dans le guide de l'AWS HealthOmics utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListSharesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour obtenir une liste de tags

L'`list-tags-for-resource`exemple suivant permet d'obtenir une liste de balises pour un flux de travail avec un identifiant1234567.

```
aws omics list-tags-for-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567
```

Sortie :

```
{
    "tags": {
        "department": "analytics"
    }
}
```

Pour plus d'informations, consultez la section Marquage des ressources dans Amazon Omics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-variant-import-jobs`

L'exemple de code suivant montre comment utiliser`list-variant-import-jobs`.

AWS CLI

Pour obtenir une liste des variantes des tâches d'importation

L'`list-variant-import-jobs`exemple suivant permet d'obtenir une liste de tâches d'importation de variantes.

```
aws omics list-variant-import-jobs
```

Sortie :

```
{
    "variantImportJobs": [
        {
            "creationTime": "2022-11-23T22:47:02.514002Z",
            "destinationName": "my_var_store",
            "id": "69cb65d6-xmpl-4a4a-9025-4565794b684e",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-23T22:49:17.976597Z"
        },
        {
            "creationTime": "2022-11-23T22:42:50.037812Z",
            "destinationName": "my_var_store",
            "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-23T22:45:26.009880Z"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListVariantImportJobsà la section Référence des AWS CLI commandes.

`list-variant-stores`

L'exemple de code suivant montre comment utiliser`list-variant-stores`.

AWS CLI

Pour obtenir une liste des variantes de magasins

L'`list-variant-stores`exemple suivant permet d'obtenir une liste de magasins de variantes.

```
aws omics list-variant-stores
```

Sortie :

```
{
    "variantStores": [
        {
            "creationTime": "2022-11-23T22:09:07.534499Z",
            "id": "02dexmplcfdd",
            "name": "my_var_store",
            "reference": {
                "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
            },
            "status": "CREATING",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
my_var_store",
            "storeSizeBytes": 0,
            "updateTime": "2022-11-23T22:09:24.931711Z"
        },
        {
            "creationTime": "2022-09-23T23:00:09.140265Z",
            "id": "8777xmpl1a24",
            "name": "myvstore0",
            "status": "ACTIVE",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
myvstore0",
```

```
            "storeSizeBytes": 0,
            "updateTime": "2022-09-23T23:03:26.013220Z"
        }
    ]
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListVariantStoresà la section Référence des AWS CLI commandes.

`list-workflows`

L'exemple de code suivant montre comment utiliser`list-workflows`.

AWS CLI

Pour obtenir une liste des flux de travail

L'`list-workflows`exemple suivant permet d'obtenir une liste de flux de travail.

```
aws omics list-workflows
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
            "creationTime": "2022-09-23T23:08:22.041227Z",
            "digest": "nSCNo/qMWFxmplXpUdokXJnwgneOaxyyc2YOxVxrJTE=",
            "id": "1234567",
            "name": "my-wkflow-0",
            "status": "ACTIVE",
            "type": "PRIVATE"
        },
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
            "creationTime": "2022-11-30T22:33:16.225368Z",
            "digest":
 "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
            "id": "1234567",
```

```
            "name": "cram-converter",
            "status": "ACTIVE",
            "type": "PRIVATE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création de flux de travail privés dans le guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListWorkflowsà la section Référence des AWS CLI commandes.

`start-annotation-import-job`

L'exemple de code suivant montre comment utiliser`start-annotation-import-job`.

AWS CLI

Pour importer des annotations

L'`start-annotation-import-job`exemple suivant importe des annotations depuis Amazon S3.

```
aws omics start-annotation-import-job \
    --destination-name tsv_ann_store \
    --no-run-left-normalization \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
    --items source=s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz
```

Sortie :

```
{
    "jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartAnnotationImportJobà la section Référence des AWS CLI commandes.

start-read-set-activation-job

L'exemple de code suivant montre comment utiliserstart-read-set-activation-job.

AWS CLI

Pour activer un ensemble de lecture archivé

L'start-read-set-activation-jobexemple suivant active deux ensembles de lecture.

```
aws omics start-read-set-activation-job \
    --sequence-store-id 1234567890 \
    --sources readSetId=1234567890 readSetId=1234567890
```

Sortie :

```
{
    "creationTime": "2022-12-06T22:35:10.100Z",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartReadSetActivationJobà la section Référence des AWS CLI commandes.

start-read-set-export-job

L'exemple de code suivant montre comment utiliserstart-read-set-export-job.

AWS CLI

Pour exporter un ensemble de lectures

L'start-read-set-export-jobexemple suivant exporte deux ensembles de lecture vers Amazon S3.

```
aws omics start-read-set-export-job \
    --sequence-store-id 1234567890 \
```

```
        --sources readSetId=1234567890 readSetId=1234567890 \
        --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ
\
        --destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

Sortie :

```
{
    "creationTime": "2022-12-06T22:37:18.612Z",
    "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartReadSetExportJobà la section Référence des AWS CLI commandes.

`start-read-set-import-job`

L'exemple de code suivant montre comment utiliser`start-read-set-import-job`.

AWS CLI

Pour importer un ensemble de lectures

L'`start-read-set-import-job`exemple suivant importe un ensemble de lectures.

```
aws omics start-read-set-import-job \
    --sequence-store-id 1234567890 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ \
    --sources file://readset-sources.json
```

readset-sources.json est un document JSON dont le contenu est le suivant.

```
[
```

```
    {
        "sourceFiles":
        {
            "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
        },
        "sourceFileType": "BAM",
        "subjectId": "bam-subject",
        "sampleId": "bam-sample",
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
        "name": "HG00100"
    }
]
```

Sortie :

```
{
    "creationTime": "2022-11-23T01:36:38.158Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartReadSetImportJobà la section Référence des AWS CLI commandes.

`start-reference-import-job`

L'exemple de code suivant montre comment utiliser`start-reference-import-job`.

AWS CLI

Pour importer un génome de référence

L'`start-reference-import-job`exemple suivant importe un génome de référence depuis Amazon S3.

```
aws omics start-reference-import-job \
    --reference-store-id 1234567890 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
    --sources sourceFile=s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta,name=assembly-38
```

Sortie :

```
{
    "creationTime": "2022-11-22T22:25:41.124Z",
    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
    "status": "SUBMITTED"
}
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartReferenceImportJobà la section Référence des AWS CLI commandes.

`start-run`

L'exemple de code suivant montre comment utiliser`start-run`.

AWS CLI

Pour exécuter un flux de travail

L'`start-run`exemple suivant exécute un flux de travail avec ID`1234567`.

```
aws omics start-run \
    --workflow-id 1234567 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
    --name 'cram-to-bam' \
    --output-uri s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/ \
    --run-group-id 1234567 \
    --priority 1 \
```

```
    --storage-capacity 10 \
    --log-level ALL \
    --parameters file://workflow-inputs.json
```

workflow-inputs.json est un document JSON dont le contenu est le suivant.

```
{
    "sample_name": "NA12878",
    "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram",
    "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
    "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
    "ref_fasta_index": "omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta.fai"
}
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "id": "1234567",
    "status": "PENDING",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Démarrage d'une course dans le guide de AWS HealthOmics l'utilisateur.

Pour charger des fichiers source depuis Amazon Omics

Vous pouvez également charger des fichiers source depuis le stockage Amazon Omics, en utilisant des fichiers spécifiques au service URIs. L'exemple de fichier workflow-inputs.json suivant utilise Amazon URIs Omics pour les ensembles de lecture et les sources génomiques de référence.

```
{
    "sample_name": "NA12878",
    "input_cram": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/
readSet/1234567890/source1",
    "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
```

```
    "ref_fasta": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/
reference/1234567890",
    "ref_fasta_index": "omics://123456789012.storage.us-
west-2.amazonaws.com/1234567890/reference/1234567890/index"
}
```

- Pour plus de détails sur l'API, reportez-vous StartRunà la section Référence des AWS CLI commandes.

`start-variant-import-job`

L'exemple de code suivant montre comment utiliser`start-variant-import-job`.

AWS CLI

Pour importer un fichier de variantes

L'`start-variant-import-job`exemple suivant importe un fichier de variante du format VCF.

```
aws omics start-variant-import-job \
    --destination-name my_var_store \
    --no-run-left-normalization  \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
    --items source=s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.known_indels.vcf.gz
```

Sortie :

```
{
    "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartVariantImportJobà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute une `department` balise à un flux de travail avec un identifiant`1234567`.

```
aws omics tag-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \
    --tags department=analytics
```

Pour plus d'informations, consultez la section Marquage des ressources dans Amazon Omics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime la `department` balise d'un flux de travail.

```
aws omics untag-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \
    --tag-keys department
```

Pour plus d'informations, consultez Omics Storage dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-annotation-store`

L'exemple de code suivant montre comment utiliser`update-annotation-store`.

AWS CLI

Pour mettre à jour un magasin d'annotations

L'`update-annotation-store`exemple suivant met à jour la description d'un magasin d'annotations nommé`my_vcf_store`.

```
aws omics update-annotation-store \
    --name my_vcf_store \
    --description "VCF annotation store"
```

Sortie :

```
{
    "creationTime": "2022-12-05T18:00:56.101860Z",
    "description": "VCF annotation store",
    "id": "bd6axmpl2444",
    "name": "my_vcf_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "ACTIVE",
    "storeFormat": "VCF",
    "updateTime": "2022-12-05T18:13:16.100051Z"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateAnnotationStoreà la section Référence des AWS CLI commandes.

`update-run-group`

L'exemple de code suivant montre comment utiliser`update-run-group`.

AWS CLI

Pour mettre à jour un groupe de course

L'`update-run-group`exemple suivant met à jour les paramètres d'un groupe d'exécution avec id`1234567`.

```
aws omics update-run-group \
    --id 1234567 \
    --max-cpus 10
```

Sortie :

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "creationTime": "2022-12-01T00:58:42.915219Z",
    "id": "1234567",
    "maxCpus": 10,
    "maxDuration": 600,
    "name": "cram-convert",
    "tags": {}
}
```

Pour plus d'informations, consultez Omics Workflows dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateRunGroupà la section Référence des AWS CLI commandes.

`update-variant-store`

L'exemple de code suivant montre comment utiliser`update-variant-store`.

AWS CLI

Pour mettre à jour un magasin de variantes

L'`update-variant-store`exemple suivant met à jour la description d'un magasin de variantes nommé`my_var_store`.

```
aws omics update-variant-store \
    --name my_var_store \
    --description "variant store"
```

Sortie :

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "description": "variant store",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "ACTIVE",
    "updateTime": "2022-12-05T18:23:37.686402Z"
}
```

Pour plus d'informations, consultez Omics Analytics dans le manuel Amazon Omics Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateVariantStoreà la section Référence des AWS CLI commandes.

`update-workflow`

L'exemple de code suivant montre comment utiliser`update-workflow`.

AWS CLI

Pour mettre à jour un flux de travail

L'`update-workflow`exemple suivant met à jour la description d'un flux de travail avec ID`1234567`.

```
aws omics update-workflow \
    --id 1234567 \
    --description "copy workflow"
```

Pour plus d'informations, voir Création ou mise à jour d'un flux de travail dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UpdateWorkflowà la section Référence des AWS CLI commandes.

`upload-read-set-part`

L'exemple de code suivant montre comment utiliser`upload-read-set-part`.

AWS CLI

Pour télécharger une partie du kit de lecture.

L'`upload-read-set-part`exemple suivant télécharge une partie spécifiée d'un ensemble de lectures.

```
aws omics upload-read-set-part \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --part-source SOURCE1 \
    --part-number 1 \
    --payload /path/to/file/read_1_part_1.fastq.gz
```

Sortie :

```
{
    "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

Pour plus d'informations, reportez-vous à la section Téléchargement direct vers un magasin de séquences dans le Guide de AWS HealthOmics l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous UploadReadSetPartà la section Référence des AWS CLI commandes.

Exemples d'IAM utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with IAM.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-client-id-to-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`add-client-id-to-open-id-connect-provider`.

AWS CLI

Pour ajouter un ID client (audience) à un fournisseur Open-ID Connect (OIDC)

La commande `add-client-id-to-open-id-connect-provider` suivante ajoute l'ID client `my-application-ID` au fournisseur OIDC nommé `server.example.com`.

```
aws iam add-client-id-to-open-id-connect-provider \
    --client-id my-application-ID \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Cette commande ne produit aucun résultat.

Pour créer un fournisseur OIDC, utilisez la commande `create-open-id-connect-provider`.

Pour plus d'informations, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AddClientIdToOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`add-role-to-instance-profile`

L'exemple de code suivant montre comment utiliser`add-role-to-instance-profile`.

AWS CLI

Pour ajouter un rôle à un profil d'instance

La commande `add-role-to-instance-profile` suivante ajoute le rôle nommé `S3Access` au profil d'instance nommé `Webserver`.

```
aws iam add-role-to-instance-profile \
    --role-name S3Access \
    --instance-profile-name Webserver
```

Cette commande ne produit aucun résultat.

Pour créer un profil d'instance, utilisez la commande `create-instance-profile`.

Pour plus d'informations, consultez la section Utilisation d'un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AddRoleToInstanceProfileà la section Référence des AWS CLI commandes.

`add-user-to-group`

L'exemple de code suivant montre comment utiliser`add-user-to-group`.

AWS CLI

Pour ajouter un utilisateur à un groupe IAM

La commande `add-user-to-group` suivante ajoute un utilisateur IAM nommé `Bob` au groupe IAM nommé `Admins`.

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Ajout et suppression d'utilisateurs dans un groupe IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AddUserToGroupà la section Référence des AWS CLI commandes.

`attach-group-policy`

L'exemple de code suivant montre comment utiliser`attach-group-policy`.

AWS CLI

Pour attacher une politique gérée à un groupe IAM

La `attach-group-policy` commande suivante associe la politique AWS gérée nommée `ReadOnlyAccess` au groupe IAM nomméFinance.

```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --group-name Finance
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Politiques gérées et politiques en ligne dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AttachGroupPolicyà la section Référence des AWS CLI commandes.

`attach-role-policy`

L'exemple de code suivant montre comment utiliserattach-role-policy.

AWS CLI

Pour attacher une politique gérée à un rôle IAM

La `attach-role-policy` commande suivante associe la politique AWS gérée nommée `ReadOnlyAccess` au rôle IAM nomméReadOnlyRole.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Politiques gérées et politiques en ligne dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AttachRolePolicyà la section Référence des AWS CLI commandes.

`attach-user-policy`

L'exemple de code suivant montre comment utiliser`attach-user-policy`.

AWS CLI

Pour attacher une politique gérée à un utilisateur IAM

La `attach-user-policy` commande suivante associe la politique AWS gérée nommée `AdministratorAccess` à l'utilisateur IAM nomméAlice.

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Politiques gérées et politiques en ligne dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AttachUserPolicyà la section Référence des AWS CLI commandes.

`change-password`

L'exemple de code suivant montre comment utiliserchange-password.

AWS CLI

Pour changer le mot de passe de votre utilisateur IAM

Pour modifier le mot de passe de votre utilisateur IAM, nous vous recommandons d'utiliser le paramètre `--cli-input-json` pour transmettre un fichier JSON contenant vos anciens et nouveaux mots de passe. Grâce à cette méthode, vous pouvez utiliser des mots de passe forts contenant des caractères non alphanumériques. Il peut être difficile d'utiliser des mots de passe contenant des caractères non alphanumériques lorsque vous les transmettez en tant que paramètres de ligne de commande. Pour utiliser le paramètre `--cli-input-json`, commencez par utiliser la commande `change-password` avec le paramètre `--generate-cli-skeleton`, comme dans l'exemple suivant.

```
aws iam change-password \
```

```
    --generate-cli-skeleton > change-password.json
```

La commande précédente crée un fichier JSON appelé change-password.json que vous pouvez utiliser pour renseigner vos anciens et nouveaux mots de passe. Par exemple, le fichier peut ressembler à ce qui suit.

```
{
    "OldPassword": "3s0K_;xh4~8XXI",
    "NewPassword": "]35d/{pB9Fo9wJ"
}
```

Ensuite, pour modifier votre mot de passe, réutilisez la commande `change-password`, en passant cette fois le paramètre `--cli-input-json` pour spécifier votre fichier JSON. La commande `change-password` suivante utilise le paramètre `--cli-input-json` avec un fichier JSON appelé change-password.json.

```
aws iam change-password \
    --cli-input-json file://change-password.json
```

Cette commande ne produit aucun résultat.

Cette commande ne peut être appelée que par les utilisateurs IAM. Si cette commande est appelée à l'aide des informations d'identification du AWS compte (root), elle renvoie une `InvalidUserType` erreur.

Pour plus d'informations, consultez Comment un utilisateur IAM modifie son propre mot de passe dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ChangePasswordà la section Référence des AWS CLI commandes.

`create-access-key`

L'exemple de code suivant montre comment utiliser`create-access-key`.

AWS CLI

Pour créer une clé d'accès pour un utilisateur IAM

La commande `create-access-key` suivante créer une clé d'accès (un ID de clé d'accès et une clé d'accès secrète) pour l'utilisateur IAM nommé Bob.

```
aws iam create-access-key \
    --user-name Bob
```

Sortie :

```
{
    "AccessKey": {
        "UserName": "Bob",
        "Status": "Active",
        "CreateDate": "2015-03-09T18:39:23.411Z",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

Conservez votre clé d'accès secrète dans un emplacement sécurisé. Si elle est perdue, elle ne peut pas être récupérée et vous devez créer une nouvelle clé.

Pour de plus amples informations, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateAccessKeyà la section Référence des AWS CLI commandes.

`create-account-alias`

L'exemple de code suivant montre comment utiliser`create-account-alias`.

AWS CLI

Pour créer un alias de compte

La `create-account-alias` commande suivante crée l'alias `examplecorp` de votre AWS compte.

```
aws iam create-account-alias \
    --account-alias examplecorp
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Votre identifiant de AWS compte et son alias dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateAccountAliasà la section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour créer un groupe IAM

La commande `create-group` suivante crée un groupe IAM nommé `Admins`.

```
aws iam create-group \
    --group-name Admins
```

Sortie :

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-03-09T20:30:24.940Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    }
}
```

Pour plus d’informations, consultez la section Création de groupes d’utilisateurs IAM dans le Guide de l’utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`create-instance-profile`

L'exemple de code suivant montre comment utiliser`create-instance-profile`.

AWS CLI

Pour créer un profil d’instance

La commande `create-instance-profile` suivante crée un profil d'instance nommé `Webserver`.

```
aws iam create-instance-profile \
    --instance-profile-name Webserver
```

Sortie :

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJMBYC7DLSPEXAMPLE",
        "Roles": [],
        "CreateDate": "2015-03-09T20:33:19.626Z",
        "InstanceProfileName": "Webserver",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"
    }
}
```

Pour ajouter un rôle à un profil d'instance, utilisez la commande `add-role-to-instance-profile`.

Pour plus d'informations, consultez la section Utilisation d'un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateInstanceProfileà la section Référence des AWS CLI commandes.

`create-login-profile`

L'exemple de code suivant montre comment utiliser`create-login-profile`.

AWS CLI

Pour générer le mot de passe d'un utilisateur IAM

Pour générer le mot de passe d'un utilisateur IAM, nous vous recommandons d'utiliser le paramètre `--cli-input-json` pour transmettre un fichier JSON contenant le mot de passe. Grâce à cette méthode, vous pouvez générer un mot de passe fort contenant des caractères non

alphanumériques. Il peut être difficile de générer un mot de passe contenant des caractères non alphanumériques lorsque vous le transmettez en tant que paramètre de la ligne de commande.

Pour utiliser le paramètre `--cli-input-json`, commencez par utiliser la commande `create-login-profile` avec le paramètre `--generate-cli-skeleton`, comme dans l'exemple suivant.

```
aws iam create-login-profile \
    --generate-cli-skeleton > create-login-profile.json
```

La commande précédente crée un fichier JSON appelé create-login-profile .json que vous pouvez utiliser pour renseigner les informations d'une `create-login-profile` commande suivante. Par exemple :

```
{
    "UserName": "Bob",
    "Password": "&1-3a6u:RA0djs",
    "PasswordResetRequired": true
}
```

Ensuite, pour générer le mot de passe d'un utilisateur IAM, réutilisez la commande `create-login-profile`, en passant cette fois le paramètre `--cli-input-json` pour spécifier votre fichier JSON. La `create-login-profile` commande suivante utilise le `--cli-input-json` paramètre avec un fichier JSON appelé create-login-profile .json.

```
aws iam create-login-profile \
    --cli-input-json file://create-login-profile.json
```

Sortie :

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2015-03-10T20:55:40.274Z",
        "PasswordResetRequired": true
    }
}
```

Si le nouveau mot de passe enfreint la politique de mot de passe du compte, la commande renvoie une erreur `PasswordPolicyViolation`.

Pour modifier le mot de passe d'un utilisateur qui en possède déjà un, utilisez `update-login-profile`. Pour définir une politique de mot de passe pour le compte, utilisez la commande `update-account-password-policy`.

Si la politique de mot de passe du compte le permet, les utilisateurs IAM peuvent modifier leurs propres mots de passe à l'aide de la commande `change-password`.

Pour de plus amples informations, consultez Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateLoginProfileà la section Référence des AWS CLI commandes.

`create-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`create-open-id-connect-provider`.

AWS CLI

Pour créer un fournisseur OpenID Connect (OIDC)

Pour créer un fournisseur OpenID Connect (OIDC), nous vous recommandons d'utiliser le paramètre `--cli-input-json` pour transmettre un fichier JSON contenant les paramètres requis. Lorsque vous créez un fournisseur OIDC, vous devez transmettre l'URL du fournisseur, qui doit commencer par `https://`. Il peut être difficile de transmettre l'URL en tant que paramètre de ligne de commande, car les deux points (:) et les barres obliques (/) ont une signification particulière dans certains environnements de ligne de commande. L'utilisation du paramètre `--cli-input-json` permet de contourner cette limitation.

Pour utiliser le paramètre `--cli-input-json`, commencez par utiliser la commande `create-open-id-connect-provider` avec le paramètre `--generate-cli-skeleton`, comme dans l'exemple suivant.

```
aws iam create-open-id-connect-provider \
    --generate-cli-skeleton > create-open-id-connect-provider.json
```

La commande précédente crée un fichier JSON appelé create-open-id-connect -provider.json que vous pouvez utiliser pour renseigner les informations d'une commande suivante. `create-open-id-connect-provider` Par exemple :

```
{
    "Url": "https://server.example.com",
    "ClientIDList": [
        "example-application-ID"
    ],
    "ThumbprintList": [
        "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
    ]
}
```

Ensuite, pour créer le fournisseur OpenID Connect (OIDC), réutilisez la commande `create-open-id-connect-provider`, en passant cette fois le paramètre `--cli-input-json` pour spécifier votre fichier JSON. La `create-open-id-connect-provider` commande suivante utilise le `--cli-input-json` paramètre avec un fichier JSON appelé create-open-id-connect -provider.json.

```
aws iam create-open-id-connect-provider \
    --cli-input-json file://create-open-id-connect-provider.json
```

Sortie :

```
{
    "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

Pour obtenir des informations sur les fournisseurs OIDC, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) dans le Guide de l'utilisateur AWS IAM.

Pour plus d'informations sur l'obtention d'empreintes numériques pour un fournisseur OIDC, consultez Obtention de l'empreinte numérique pour un fournisseur d'identité OpenID Connect dans le Guide de l'utilisateur AWS IAM

- Pour plus de détails sur l'API, reportez-vous CreateOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`create-policy-version`

L'exemple de code suivant montre comment utiliser`create-policy-version`.

AWS CLI

Pour créer une nouvelle version de la politique gérée

Cet exemple crée une nouvelle version v2 de la politique IAM dont l'ARN est `arn:aws:iam::123456789012:policy/MyPolicy` et en fait la version par défaut.

```
aws iam create-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --policy-document file://NewPolicyVersion.json \
    --set-as-default
```

Sortie :

```
{
    "PolicyVersion": {
        "CreateDate": "2015-06-16T18:56:03.721Z",
        "VersionId": "v2",
        "IsDefaultVersion": true
    }
}
```

Pour plus d'informations, consultez Gestion des versions des politiques IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyVersionà la section Référence des AWS CLI commandes.

`create-policy`

L'exemple de code suivant montre comment utiliser`create-policy`.

AWS CLI

Exemple 1 : Pour créer une politique gérée par le client

La commande suivante crée une politique gérée par le client nommée `my-policy`. Le fichier `policy.json` est un document JSON dans le dossier actuel qui accorde un accès en lecture seule au dossier `shared` dans un compartiment Amazon S3 nommé `amzn-s3-demo-bucket`.

```
aws iam create-policy \
```

```
    --policy-name my-policy \
    --policy-document file://policy.json
```

Contenu de policy.json

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/shared/*"
            ]
        }
    ]
}
```

Sortie :

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "CreateDate": "2015-06-01T19:31:18.620Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::0123456789012:policy/my-policy",
        "UpdateDate": "2015-06-01T19:31:18.620Z"
    }
}
```

Pour plus d'informations sur l'utilisation de fichiers comme entrée pour les paramètres de chaîne, voir Spécifier les valeurs des paramètres pour la AWS CLI dans le guide de l'utilisateur de la AWS CLI.

Exemple 2 : Pour créer une politique gérée par le client avec une description

La commande suivante crée une politique gérée par le client nommée `my-policy` avec une description immuable.

Le fichier `policy.json` est un document JSON dans le dossier actuel qui accorde un accès à toutes les actions Pull, List et Get dans un compartiment Amazon S3 nommé `amzn-s3-demo-bucket`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --description "This policy grants access to all Put, Get, and List actions for
 amzn-s3-demo-bucket"
```

Contenu de policy.json

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

Sortie :

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
```

```
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T22:38:47+00:00",
        "UpdateDate": "2023-05-24T22:38:47+00:00"
    }
}
```

Pour en savoir plus sur les politiques basées sur l'identité, consultez Politiques basées sur l'identité et Politiques basées sur une ressource dans le Guide de l'utilisateur AWS IAM.

Exemple 3 : pour créer une politique gérée par le client avec des balises

La commande suivante crée une politique gérée par le client nommée `my-policy` avec des balises. Cet exemple utilise le paramètre `--tags` avec les balises au format JSON suivantes : `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. Le paramètre `--tags` peut également être utilisé avec des balises au format abrégé : `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

Le fichier `policy.json` est un document JSON dans le dossier actuel qui accorde un accès à toutes les actions Pull, List et Get dans un compartiment Amazon S3 nommé `amzn-s3-demo-bucket`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Contenu de policy.json

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

Sortie :

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::12345678012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T23:16:39+00:00",
        "UpdateDate": "2023-05-24T23:16:39+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
                "Key": "Location",
                "Value": "Seattle"
            {
        ]
    }
}
```

Pour plus d'informations sur les politiques de balisage, consultez Balisage des politiques gérées par le client dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyà la section Référence des AWS CLI commandes.

`create-role`

L'exemple de code suivant montre comment utiliser`create-role`.

AWS CLI

Exemple 1 : Pour créer un rôle IAM

La commande `create-role` suivante crée un rôle nommé `Test-Role` et lui attache une politique d'approbation.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

Sortie :

```
{
    "Role": {
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "CreateDate": "2013-06-07T20:43:32.821Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

La politique d'approbation est définie sous la forme d'un document JSON dans le fichier Test-Role-Trust-Policy.json. (Le nom et l'extension du fichier n'ont aucune importance.) La politique d'approbation doit spécifier un principal.

Pour attacher une politique des autorisations à un rôle, utilisez la commande `put-role-policy`.

Pour plus d'informations, consultez Création de rôles IAM dans le Guide de l'utilisateur AWS IAM.

Exemple 2 : Pour créer un rôle IAM avec une durée de session maximale spécifiée

La commande `create-role` suivante crée un rôle nommé `Test-Role` et définit une durée de session maximale de 7 200 secondes (2 heures).

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
```

```
    --max-session-duration 7200
```

Sortie :

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:role/Test-Role",
        "CreateDate": "2023-05-24T23:50:25+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::12345678012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez la section Modification de la durée maximale de session (AWS API) d'un rôle dans le guide de l'utilisateur AWS IAM.

Exemple 3 : Pour créer un rôle IAM avec des balises

La commande suivante crée un rôle IAM `Test-Role` avec des balises. Cet exemple utilise l'indicateur de paramètre `--tags` avec les balises au format JSON suivantes : `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. L'indicateur `--tags` peut également être utilisé avec des balises au format raccourci : `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
```

```
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Sortie :

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role",
        "CreateDate": "2023-05-25T23:29:41+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

Pour plus d'informations, veuillez consulter Étiquette de rôles IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateRoleà la section Référence des AWS CLI commandes.

`create-saml-provider`

L'exemple de code suivant montre comment utiliser`create-saml-provider`.

AWS CLI

Pour créer un fournisseur SAML

Cet exemple crée un nouveau fournisseur SAML dans IAM nommé `MySAMLProvider`. Il est décrit par le document de métadonnées SAML présent dans le fichier `SAMLMetaData.xml`.

```
aws iam create-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --name MySAMLProvider
```

Sortie :

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

Pour plus d'informations, consultez Création de fournisseurs d'identité SAML IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, voir Create SAMLProvider in AWS CLI Command Reference.

`create-service-linked-role`

L'exemple de code suivant montre comment utiliser`create-service-linked-role`.

AWS CLI

Pour créer un rôle lié à un service

L'`create-service-linked-role`exemple suivant crée un rôle lié à un service pour le AWS service spécifié et y joint la description spécifiée.

```
aws iam create-service-linked-role \
    --aws-service-name lex.amazonaws.com \
    --description "My service-linked role to support Lex"
```

Sortie :

```
{
    "Role": {
        "Path": "/aws-service-role/lex.amazonaws.com/",
        "RoleName": "AWSServiceRoleForLexBots",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots",
        "CreateDate": "2019-04-17T20:34:14+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "lex.amazonaws.com"
                        ]
                    }
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez Utilisation des rôles liés à un service dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateServiceLinkedRoleà la section Référence des AWS CLI commandes.

`create-service-specific-credential`

L'exemple de code suivant montre comment utiliser`create-service-specific-credential`.

AWS CLI

Création d'un ensemble d'informations d'identification spécifiques au service pour un utilisateur

L'`create-service-specific-credential`exemple suivant crée un nom d'utilisateur et un mot de passe qui ne peuvent être utilisés que pour accéder au service configuré.

```
aws iam create-service-specific-credential \
    --user-name sofia \
    --service-name codecommit.amazonaws.com
```

Sortie :

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "k1zPZM6uVxMQ3oxqgoYlNuJPyRTZ1vREs76zTQE3eJk=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

Pour plus d'informations, consultez la section Créer des informations d'identification Git pour les connexions HTTPS CodeCommit dans le Guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous CreateServiceSpecificCredentialà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Exemple 1 : Pour créer un utilisateur IAM

La commande `create-user` suivante crée un utilisateur IAM nommé Bob dans le compte actuel.

```
aws iam create-user \
    --user-name Bob
```

Sortie :

```
{
    "User": {
```

```
        "UserName": "Bob",
        "Path": "/",
        "CreateDate": "2023-06-08T03:20:41.270Z",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
}
```

Pour plus d'informations, consultez la section Création d'un utilisateur IAM dans votre AWS compte dans le guide de l'utilisateur AWS IAM.

Exemple 2 : Pour créer un utilisateur IAM à un chemin spécifié

La commande `create-user` suivante crée un utilisateur IAM nommé Bob au chemin spécifié.

```
aws iam create-user \
    --user-name Bob \
    --path /division_abc/subdivision_xyz/
```

Sortie :

```
{
    "User": {
        "Path": "/division_abc/subdivision_xyz/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/division_abc/subdivision_xyz/Bob",
        "CreateDate": "2023-05-24T18:20:17+00:00"
    }
}
```

Pour plus d'informations, consultez Identifiants IAM dans le Guide de l'utilisateur AWS IAM.

Exemple 3 : Pour créer un utilisateur IAM avec des balises

La commande `create-user` suivante crée un utilisateur IAM nommé Bob avec des balises. Cet exemple utilise l'indicateur de paramètre `--tags` avec les balises au format JSON suivantes : `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. L'indicateur `--tags` peut également être utilisé avec des balises au format raccourci : `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-user \
    --user-name Bob \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Sortie :

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-25T17:14:21+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

Pour plus d'informations, consultez Étiquette d'utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

Exemple 3 : Pour créer un utilisateur IAM avec une limite des autorisations définie

La `create-user` commande suivante crée un utilisateur IAM dont le nom correspond à la limite Bob d'autorisations d'FullAccessAmazonS3.

```
aws iam create-user \
    --user-name Bob \
    --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

Sortie :

```
{
```

```
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-24T17:50:53+00:00",
        "PermissionsBoundary": {
        "PermissionsBoundaryType": "Policy",
        "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
        }
    }
}
```

Pour plus d'informations, consultez Limites d'autorisations pour les entités IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`create-virtual-mfa-device`

L'exemple de code suivant montre comment utiliser`create-virtual-mfa-device`.

AWS CLI

Pour créer un dispositif MFA virtuel

Cet exemple crée un dispositif MFA virtuel appelé `BobsMFADevice`. Il crée un fichier, appelé `QRCode.png`, qui contient les informations d'amorçage et le place dans le répertoire `C:/`. La méthode d'amorçage utilisée dans cet exemple est `QRCodePNG`.

```
aws iam create-virtual-mfa-device \
    --virtual-mfa-device-name BobsMFADevice \
    --outfile C:/QRCode.png \
    --bootstrap-method QRCodePNG
```

Sortie :

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
}
```

Pour plus d'informations, consultez Utilisation de l'authentification multifactorielle (MFA) dans AWS dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateVirtualMfaDeviceà la section Référence des AWS CLI commandes.

`deactivate-mfa-device`

L'exemple de code suivant montre comment utiliser`deactivate-mfa-device`.

AWS CLI

Pour désactiver un dispositif MFA

Cette commande désactive le dispositif MFA virtuel, dont l'ARN est `arn:aws:iam::210987654321:mfa/BobsMFADevice`, associé à l'utilisateur Bob.

```
aws iam deactivate-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Utilisation de l'authentification multifactorielle (MFA) dans AWS dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeactivateMfaDeviceà la section Référence des AWS CLI commandes.

`decode-authorization-message`

L'exemple de code suivant montre comment utiliser`decode-authorization-message`.

AWS CLI

Pour décoder un message d'échec d'autorisation

L'`decode-authorization-message`exemple suivant décode le message renvoyé par la EC2 console lors de la tentative de lancement d'une instance sans les autorisations requises.

```
aws sts decode-authorization-message \
```

```
    --encoded-message lxzA8VEjEvu-s0TTt3PgYCXik9YakOqsrFJGRZR98xNcyWAxwRq14xIvd-
npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-
mWR9hVYdvX-0zKgVOWF9pWjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpu0IjxDjzf22PF3SOE3XvIQ-
_PEO0aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8lsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX
Xx9AYAAIr6bhcis7C__bZh4dlAAWooHFGKgfoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

La sortie est formatée sous la forme d'une chaîne de texte JSON d'une seule ligne que vous pouvez analyser avec n'importe quel processeur de texte JSON.

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements
\":{\"items\":[]},\"failures\":{\"items\":[]},\"context\":{\"principal
\":{\"id\":\"AIDAV3ZUEFP6J7GY7O6LO\",\"name\":\"chain-user\",\"arn\":
\"arn:aws:iam::403299380220:user/chain-user\"},\"action\":\"ec2:RunInstances\",
\"resource\":\"arn:aws:ec2:us-east-2:403299380220:instance/*\",\"conditions\":
{\"items\":[{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value
\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value
\":\"instance/*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value
\":\"403299380220\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":
[{\"value\":\"us-east-2b\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items
\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values
\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values
\":{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",
\"values\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values
\":{\"items\":[{\"value\":\"us-east-2\"}]}},{\"key\":\"aws:Service\",\"values
\":{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":
{\"items\":[{\"value\":\"*\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":
[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":
[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-2\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":
\"arn:aws:ec2:us-east-2:403299380220:instance/*\"}]}}]}}}"
}
```

Pour plus d'informations, consultez Comment décoder un message d'échec d'autorisation après avoir reçu une erreur « UnauthorizedOperation » lors du lancement d'une EC2 instance ? dans AWS Re:post.

- Pour plus de détails sur l'API, reportez-vous DecodeAuthorizationMessageà la section Référence des AWS CLI commandes.

`delete-access-key`

L'exemple de code suivant montre comment utiliser`delete-access-key`.

AWS CLI

Pour supprimer une clé d'accès d'un utilisateur IAM

La commande `delete-access-key` suivante supprime la clé d'accès spécifiée (un ID de clé d'accès rapide et une clé d'accès secrète) de l'utilisateur IAM nommé Bob.

```
aws iam delete-access-key \
    --access-key-id AKIDPMS9RO4H3FEXAMPLE \
    --user-name Bob
```

Cette commande ne produit aucun résultat.

Pour répertorier les clés d'accès définies pour un utilisateur IAM, utilisez la commande `list-access-keys`.

Pour de plus amples informations, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteAccessKeyà la section Référence des AWS CLI commandes.

`delete-account-alias`

L'exemple de code suivant montre comment utiliser`delete-account-alias`.

AWS CLI

Pour supprimer un alias de compte

La commande `delete-account-alias` suivante supprime les alias `mycompany` du compte actuel.

```
aws iam delete-account-alias \
    --account-alias mycompany
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Votre identifiant de AWS compte et son alias dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteAccountAliasà la section Référence des AWS CLI commandes.

`delete-account-password-policy`

L'exemple de code suivant montre comment utiliser`delete-account-password-policy`.

AWS CLI

Pour supprimer la politique de mot de passe du compte actuel

La commande `delete-account-password-policy` suivante supprime la politique de mot de passe du compte actuel.

```
aws iam delete-account-password-policy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition d'une politique de mot de passe du compte pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteAccountPasswordPolicyà la section Référence des AWS CLI commandes.

`delete-group-policy`

L'exemple de code suivant montre comment utiliser`delete-group-policy`.

AWS CLI

Pour supprimer une politique d'un groupe IAM

La commande `delete-group-policy` suivante supprime la politique nommée `ExamplePolicy` du groupe nommé `Admins`.

```
aws iam delete-group-policy \
    --group-name Admins \
    --policy-name ExamplePolicy
```

Cette commande ne produit aucun résultat.

Pour voir les politiques attachées à un groupe, utilisez la commande `list-group-policies`.

Pour plus d'informations, consultez Gestion des politiques IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupPolicyà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe IAM

La commande `delete-group` suivante supprime un groupe IAM nommé `MyTestGroup`.

```
aws iam delete-group \
    --group-name MyTestGroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un groupe d'utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`delete-instance-profile`

L'exemple de code suivant montre comment utiliser`delete-instance-profile`.

AWS CLI

Pour supprimer un profil d'instance

La commande `delete-instance-profile` suivante supprime un profil d'instance nommé `ExampleInstanceProfile`.

```
aws iam delete-instance-profile \
```

```
    --instance-profile-name ExampleInstanceProfile
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, consultez Utilisation de profils d'instance dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteInstanceProfileà la section Référence des AWS CLI commandes.

`delete-login-profile`

L'exemple de code suivant montre comment utiliser`delete-login-profile`.

AWS CLI

Pour supprimer le mot de passe d'un utilisateur IAM

La commande `delete-login-profile` suivante supprime le mot de passe de l'utilisateur IAM nommé Bob.

```
aws iam delete-login-profile \
    --user-name Bob
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, consultez Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteLoginProfileà la section Référence des AWS CLI commandes.

`delete-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`delete-open-id-connect-provider`.

AWS CLI

Pour supprimer un fournisseur d'identité IAM OpenID Connect

Cet exemple supprime le fournisseur OIDC IAM qui se connecte au fournisseur `example.oidcprovider.com`.

```
aws iam delete-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`delete-policy-version`

L'exemple de code suivant montre comment utiliser`delete-policy-version`.

AWS CLI

Pour supprimer une version d'une politique gérée

Cet exemple supprime la version identifiée `v2` de la politique dont l'ARN est `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyVersionà la section Référence des AWS CLI commandes.

`delete-policy`

L'exemple de code suivant montre comment utiliser`delete-policy`.

AWS CLI

Pour supprimer une politique IAM

Cet exemple supprime la politique dont l'ARN est `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`delete-role-permissions-boundary`

L'exemple de code suivant montre comment utiliser`delete-role-permissions-boundary`.

AWS CLI

Pour supprimer une limite des autorisations d'un rôle IAM

L'exemple `delete-role-permissions-boundary` suivant supprime la limite des autorisations pour le rôle IAM spécifié. Pour appliquer une limite des autorisations à un rôle, utilisez la commande `put-role-permissions-boundary`.

```
aws iam delete-role-permissions-boundary \
    --role-name lambda-application-role
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteRolePermissionsBoundaryà la section Référence des AWS CLI commandes.

`delete-role-policy`

L'exemple de code suivant montre comment utiliser`delete-role-policy`.

AWS CLI

Pour supprimer une politique d'un rôle IAM

La commande `delete-role-policy` suivante supprime la politique nommée `ExamplePolicy` du rôle nommé `Test-Role`.

```
aws iam delete-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modification d'un rôle dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteRolePolicyà la section Référence des AWS CLI commandes.

`delete-role`

L'exemple de code suivant montre comment utiliser`delete-role`.

AWS CLI

Pour supprimer un rôle IAM

La commande `delete-role` suivante supprime le rôle nommé `Test-Role`.

```
aws iam delete-role \
    --role-name Test-Role
```

Cette commande ne produit aucun résultat.

Pour pouvoir supprimer un rôle, vous devez le supprimer de tout profil d'instance (`remove-role-from-instance-profile`), détacher toutes les politiques gérées (`detach-role-policy`) et supprimer toutes les politiques intégrées attachées au rôle (`delete-role-policy`).

Pour plus d'informations, consultez Création de rôles IAM et Utilisation de profils d'instance dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteRoleà la section Référence des AWS CLI commandes.

`delete-saml-provider`

L'exemple de code suivant montre comment utiliser`delete-saml-provider`.

AWS CLI

Pour supprimer un fournisseur SAML

Cet exemple supprime le fournisseur IAM SAML 2.0 dont l'ARN est `arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`.

```
aws iam delete-saml-provider \
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Création de fournisseurs d'identité SAML IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, voir Supprimer SAMLProvider dans le manuel de référence des AWS CLI commandes.

`delete-server-certificate`

L'exemple de code suivant montre comment utiliser`delete-server-certificate`.

AWS CLI

Pour supprimer un certificat de serveur de votre AWS compte

La `delete-server-certificate` commande suivante supprime le certificat de serveur spécifié de votre AWS compte.

```
aws iam delete-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

Cette commande ne produit aucun résultat.

Pour répertorier les certificats de serveur disponibles dans votre AWS compte, utilisez la `list-server-certificates` commande.

Pour de plus amples informations, veuillez consulter Gestion des certificats de serveur dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteServerCertificateà la section Référence des AWS CLI commandes.

`delete-service-linked-role`

L'exemple de code suivant montre comment utiliser`delete-service-linked-role`.

AWS CLI

Pour supprimer un rôle lié à un service

L'exemple `delete-service-linked-role` suivant supprime le rôle lié à un service spécifié dont vous n'avez plus besoin. La suppression s'effectue de manière asynchrone. Vous pouvez vérifier le statut de la suppression et si elle est terminée à l'aide de la commande `get-service-linked-role-deletion-status`.

```
aws iam delete-service-linked-role \
    --role-name AWSServiceRoleForLexBots
```

Sortie :

```
{
    "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"
}
```

Pour plus d'informations, consultez Utilisation des rôles liés à un service dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceLinkedRoleà la section Référence des AWS CLI commandes.

`delete-service-specific-credential`

L'exemple de code suivant montre comment utiliser`delete-service-specific-credential`.

AWS CLI

Exemple 1 : Supprimer les informations d'identification spécifiques au service pour l'utilisateur demandeur

L'`delete-service-specific-credential`exemple suivant supprime les informations d'identification spécifiques au service spécifiées pour l'utilisateur qui fait la demande. Le `service-specific-credential-id` est fourni lorsque vous créez l'identifiant et vous pouvez le récupérer à l'aide de la `list-service-specific-credentials` commande.

```
aws iam delete-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Cette commande ne produit aucun résultat.

Exemple 2 : Supprimer les informations d'identification spécifiques à un service pour un utilisateur spécifié

L'`delete-service-specific-credential`exemple suivant supprime les informations d'identification spécifiques au service spécifiées pour l'utilisateur spécifié. Le `service-specific-credential-id` est fourni lorsque vous créez l'identifiant et vous pouvez le récupérer à l'aide de la `list-service-specific-credentials` commande.

```
aws iam delete-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer des informations d'identification Git pour les connexions HTTPS CodeCommit dans le Guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceSpecificCredentialà la section Référence des AWS CLI commandes.

`delete-signing-certificate`

L'exemple de code suivant montre comment utiliser`delete-signing-certificate`.

AWS CLI

Pour supprimer un certificat de signature d’un utilisateur IAM

La commande `delete-signing-certificate` suivante supprime le certificat de signature spécifié pour l’utilisateur IAM nommé Bob.

```
aws iam delete-signing-certificate \
    --user-name Bob \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour obtenir l'ID d'un certificat de signature, utilisez la commande `list-signing-certificates`.

Pour plus d'informations, consultez la section Gérer les certificats de signature dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteSigningCertificateà la section Référence des AWS CLI commandes.

`delete-ssh-public-key`

L'exemple de code suivant montre comment utiliser`delete-ssh-public-key`.

AWS CLI

Pour supprimer une clé publique SSH attachée à un utilisateur IAM

La `delete-ssh-public-key` commande suivante supprime la clé publique SSH spécifiée attachée à l'utilisateur IAM. `sofia`

```
aws iam delete-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les clés SSH et SSH avec CodeCommit dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteSshPublicKeyà la section Référence des AWS CLI commandes.

`delete-user-permissions-boundary`

L'exemple de code suivant montre comment utiliser`delete-user-permissions-boundary`.

AWS CLI

Pour supprimer une limite des autorisations d'un utilisateur IAM

L'exemple `delete-user-permissions-boundary` suivant supprime la limite des autorisations attachée à l'utilisateur IAM nommé `intern`. Pour appliquer une limite des autorisations à un utilisateur, utilisez la commande `put-user-permissions-boundary`.

```
aws iam delete-user-permissions-boundary \
    --user-name intern
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteUserPermissionsBoundaryà la section Référence des AWS CLI commandes.

`delete-user-policy`

L'exemple de code suivant montre comment utiliser`delete-user-policy`.

AWS CLI

Pour supprimer une politique d'un utilisateur IAM

La commande `delete-user-policy` suivante supprime la politique spécifiée de l'utilisateur IAM nommé Bob.

```
aws iam delete-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

Cette commande ne produit aucun résultat.

Pour obtenir une liste des politiques d'un utilisateur IAM, utilisez la commande `list-user-policies`.

Pour plus d'informations, consultez la section Création d'un utilisateur IAM dans votre AWS compte dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteUserPolicyà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur IAM

La commande `delete-user` suivante supprime l'utilisateur IAM nommé Bob du compte actuel.

```
aws iam delete-user \
    --user-name Bob
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un utilisateur IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`delete-virtual-mfa-device`

L'exemple de code suivant montre comment utiliser`delete-virtual-mfa-device`.

AWS CLI

Pour retirer un dispositif MFA virtuel

La commande `delete-virtual-mfa-device` suivante supprime le dispositif MFA spécifié du compte actuel.

```
aws iam delete-virtual-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Désactivation des dispositifs MFA dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteVirtualMfaDeviceà la section Référence des AWS CLI commandes.

`detach-group-policy`

L'exemple de code suivant montre comment utiliser`detach-group-policy`.

AWS CLI

Pour détacher une politique d'un groupe

Cet exemple supprime la politique gérée, dont l'ARN est `arn:aws:iam::123456789012:policy/TesterAccessPolicy`, du rôle nommé `Testers`.

```
aws iam detach-group-policy \
    --group-name Testers \
    --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des groupes IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DetachGroupPolicyà la section Référence des AWS CLI commandes.

`detach-role-policy`

L'exemple de code suivant montre comment utiliser`detach-role-policy`.

AWS CLI

Pour détacher une politique d'un rôle

Cet exemple supprime la politique gérée avec l'ARN `arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy` du rôle nommé `FedTesterRole`.

```
aws iam detach-role-policy \
    --role-name FedTesterRole \
    --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modification d'un rôle dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DetachRolePolicyà la section Référence des AWS CLI commandes.

`detach-user-policy`

L'exemple de code suivant montre comment utiliser`detach-user-policy`.

AWS CLI

Pour détacher une politique d'un utilisateur

Cet exemple supprime la politique gérée avec l'ARN `arn:aws:iam::123456789012:policy/TesterPolicy` de l'utilisateur `Bob`.

```
aws iam detach-user-policy \
    --user-name Bob \
    --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modification des autorisations pour un utilisateur IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DetachUserPolicyà la section Référence des AWS CLI commandes.

`disable-organizations-root-credentials-management`

L'exemple de code suivant montre comment utiliser`disable-organizations-root-credentials-management`.

AWS CLI

Pour désactiver RootCredentialsManagement cette fonctionnalité dans votre organisation

La `disable-organizations-root-credentials-management` commande suivante désactive la gestion des informations d'identification des utilisateurs root privilégiés sur les comptes membres de votre organisation.

```
aws iam disable-organizations-root-credentials-management
```

Sortie :

```
{
    "EnabledFeatures": [
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

Pour plus d'informations, consultez la section Centralisation de l'accès root pour les comptes des membres dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DisableOrganizationsRootCredentialsManagementà la section Référence des AWS CLI commandes.

`disable-organizations-root-sessions`

L'exemple de code suivant montre comment utiliser`disable-organizations-root-sessions`.

AWS CLI

Pour désactiver RootSessions cette fonctionnalité dans votre organisation

La `disable-organizations-root-sessions` commande suivante désactive les sessions utilisateur root pour les tâches privilégiées sur les comptes membres de votre organisation.

```
aws iam disable-organizations-root-sessions
```

Sortie :

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

Pour plus d'informations, consultez la section Centralisation de l'accès root pour les comptes des membres dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DisableOrganizationsRootSessionsà la section Référence des AWS CLI commandes.

`enable-mfa-device`

L'exemple de code suivant montre comment utiliser`enable-mfa-device`.

AWS CLI

Pour activer un dispositif MFA

Après avoir utilisé la `create-virtual-mfa-device` commande pour créer un dispositif MFA virtuel, vous pouvez attribuer le dispositif MFA à un utilisateur. L'exemple `enable-mfa-device` suivant attribue le dispositif MFA, dont le numéro de série est `arn:aws:iam::210987654321:mfa/BobsMFADevice`, à l'utilisateur Bob. La commande synchronise également le dispositif AWS en incluant les deux premiers codes en séquence à partir du dispositif MFA virtuel.

```
aws iam enable-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 789012
```

Cette commande ne produit aucun résultat.

Pour de plus d'informations, consultez Activation d'un dispositif virtuel d'authentification multifactorielle (MFA) dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous EnableMfaDeviceà la section Référence des AWS CLI commandes.

`enable-organizations-root-credentials-management`

L'exemple de code suivant montre comment utiliser`enable-organizations-root-credentials-management`.

AWS CLI

Pour activer RootCredentialsManagement cette fonctionnalité dans votre organisation

La `enable-organizations-root-credentials-management` commande suivante permet de gérer les informations d'identification des utilisateurs root privilégiés sur les comptes membres de votre organisation.

```
aws iam enable-organizations-root-credentials-management
```

Sortie :

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

Pour plus d'informations, consultez la section Centralisation de l'accès root pour les comptes des membres dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous EnableOrganizationsRootCredentialsManagementà la section Référence des AWS CLI commandes.

`enable-organizations-root-sessions`

L'exemple de code suivant montre comment utiliser`enable-organizations-root-sessions`.

AWS CLI

Pour activer RootSessions cette fonctionnalité dans votre organisation

La `enable-organizations-root-sessions` commande suivante permet au compte de gestion ou à l'administrateur délégué d'effectuer des tâches privilégiées sur les comptes des membres de votre organisation.

```
aws iam enable-organizations-root-sessions
```

Sortie :

```
{
    "EnabledFeatures": [
        "RootSessions"
```

```
    ]
    "OrganizationId": "o-aa111bb222"
}
```

Pour plus d'informations, consultez la section Centralisation de l'accès root pour les comptes des membres dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous EnableOrganizationsRootSessionsà la section Référence des AWS CLI commandes.

`generate-credential-report`

L'exemple de code suivant montre comment utiliser`generate-credential-report`.

AWS CLI

Pour générer un rapport sur les informations d'identification

L'exemple suivant tente de générer un rapport d'identification pour le AWS compte.

```
aws iam generate-credential-report
```

Sortie :

```
{
    "State":  "STARTED",
    "Description": "No report exists. Starting a new report generation task"
}
```

Pour plus d'informations, consultez la section Obtenir des rapports d'identification pour votre AWS compte dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GenerateCredentialReportà la section Référence des AWS CLI commandes.

`generate-organizations-access-report`

L'exemple de code suivant montre comment utiliser`generate-organizations-access-report`.

AWS CLI

Exemple 1 : pour générer un rapport d'accès pour un root dans une organisation

L'`generate-organizations-access-report`exemple suivant lance une tâche en arrière-plan afin de créer un rapport d'accès pour la racine spécifiée dans une organisation. Vous pouvez afficher le rapport une fois qu'il a été créé en exécutant la `get-organizations-access-report` commande.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb
```

Sortie :

```
{
    "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"
}
```

Exemple 2 : pour générer un rapport d'accès pour un compte dans une organisation

L'`generate-organizations-access-report`exemple suivant lance une tâche en arrière-plan pour créer un rapport d'accès pour l'ID de compte `123456789012` dans l'organisation`o-4fxmplt198`. Vous pouvez afficher le rapport une fois qu'il a été créé en exécutant la `get-organizations-access-report` commande.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb/123456789012
```

Sortie :

```
{
    "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

Exemple 3 : Pour générer un rapport d'accès pour un compte dans une unité organisationnelle d'une organisation

L'`generate-organizations-access-report`exemple suivant lance une tâche en arrière-plan pour créer un rapport d'accès pour l'ID de compte `234567890123` dans une unité ou`-c3xb-lmu7j2yg` organisationnelle de l'organisation`o-4fxmplt198`. Vous pouvez afficher le rapport une fois qu'il a été créé en exécutant la `get-organizations-access-report` commande.

```
aws iam generate-organizations-access-report \
```

```
    --entity-path o-4fxmplt198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

Sortie :

```
{
    "JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

Pour obtenir des informations sur les racines et les unités organisationnelles de votre organisation, utilisez les `organizations list-organizational-units-for-parent` commandes `organizations list-roots` et.

Pour plus d'informations, voir Affiner les autorisations lors de AWS l'utilisation des dernières informations consultées dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GenerateOrganizationsAccessReportà la section Référence des AWS CLI commandes.

`generate-service-last-accessed-details`

L'exemple de code suivant montre comment utiliser`generate-service-last-accessed-details`.

AWS CLI

Exemple 1 : pour générer un rapport d'accès aux services pour une politique personnalisée

L'exemple `generate-service-last-accessed-details` suivant lance une tâche en arrière-plan pour générer un rapport répertoriant les services auxquels accèdent les utilisateurs IAM et les autres entités avec une politique personnalisée nommée `intern-boundary`. Vous pouvez afficher le rapport une fois qu'il a été créé en exécutant la commande `get-service-last-accessed-details`.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::123456789012:policy/intern-boundary
```

Sortie :

```
{
    "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
```

```
}
```

Exemple 2 : pour générer un rapport d'accès aux services pour la AdministratorAccess politique AWS gérée

L'`generate-service-last-accessed-details`exemple suivant lance une tâche en arrière-plan pour générer un rapport répertoriant les services auxquels accèdent les utilisateurs IAM et les autres entités avec la `AdministratorAccess` politique AWS gérée. Vous pouvez afficher le rapport une fois qu'il a été créé en exécutant la commande `get-service-last-accessed-details`.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::aws:policy/AdministratorAccess
```

Sortie :

```
{
    "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"
}
```

Pour plus d'informations, voir Affiner les autorisations lors de AWS l'utilisation des dernières informations consultées dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GenerateServiceLastAccessedDetailsà la section Référence des AWS CLI commandes.

`get-access-key-last-used`

L'exemple de code suivant montre comment utiliser`get-access-key-last-used`.

AWS CLI

Pour récupérer des informations relatives au moment où la clé d'accès spécifiée a été utilisée pour la dernière fois

L'exemple suivant récupère des informations relatives au moment où la clé d'accès `ABCDEXAMPLE` a été utilisée pour la dernière fois.

```
aws iam get-access-key-last-used \
    --access-key-id ABCDEXAMPLE
```

Sortie :

```
{
    "UserName":  "Bob",
    "AccessKeyLastUsed": {
        "Region": "us-east-1",
        "ServiceName": "iam",
        "LastUsedDate": "2015-06-16T22:45:00Z"
    }
}
```

Pour de plus amples informations, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAccessKeyLastUsedà la section Référence des AWS CLI commandes.

`get-account-authorization-details`

L'exemple de code suivant montre comment utiliser`get-account-authorization-details`.

AWS CLI

Pour répertorier les utilisateurs, les groupes, les rôles et les politiques IAM d'un AWS compte

La `get-account-authorization-details` commande suivante renvoie des informations sur tous les utilisateurs, groupes, rôles et politiques IAM du AWS compte.

```
aws iam get-account-authorization-details
```

Sortie :

```
{
    "RoleDetailList": [
        {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
```

```
                    "Service": "ec2.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "RoleId": "AROA1234567890EXAMPLE",
    "CreateDate": "2014-07-30T17:09:20Z",
    "InstanceProfileList": [
        {
            "InstanceProfileId": "AIPA1234567890EXAMPLE",
            "Roles": [
                {
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Sid": "",
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    },
                    "RoleId": "AROA1234567890EXAMPLE",
                    "CreateDate": "2014-07-30T17:09:20Z",
                    "RoleName": "EC2role",
                    "Path": "/",
                    "Arn": "arn:aws:iam::123456789012:role/EC2role"
                }
            ],
            "CreateDate": "2014-07-30T17:09:20Z",
            "InstanceProfileName": "EC2role",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
        }
    ],
    "RoleName": "EC2role",
    "Path": "/",
    "AttachedManagedPolicies": [
        {
            "PolicyName": "AmazonS3FullAccess",
```

```
                    "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
                },
                {
                    "PolicyName": "AmazonDynamoDBFullAccess",
                    "PolicyArn": "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess"
                }
            ],
            "RoleLastUsed": {
                "Region": "us-west-2",
                "LastUsedDate": "2019-11-13T17:30:00Z"
            },
            "RolePolicyList": [],
            "Arn": "arn:aws:iam::123456789012:role/EC2role"
        }
    ],
    "GroupDetailList": [
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": {
                "PolicyName": "AdministratorAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
            },
            "GroupName": "Admins",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "CreateDate": "2013-10-14T18:32:24Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": {
                "PolicyName": "PowerUserAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
            },
            "GroupName": "Dev",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Dev",
            "CreateDate": "2013-10-14T18:33:55Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": [],
            "GroupName": "Finance",
```

```
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Finance",
            "CreateDate": "2013-10-14T18:57:48Z",
            "GroupPolicyList": [
                {
                    "PolicyName": "policygen-201310141157",
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Action": "aws-portal:*",
                                "Sid": "Stmt1381777017000",
                                "Resource": "*",
                                "Effect": "Allow"
                            }
                        ]
                    }
                }
            ]
        }
    ],
    "UserDetailList": [
        {
            "UserName": "Alice",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:24Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Alice"
        },
        {
            "UserName": "Bob",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:25Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [
                {
                    "PolicyName": "DenyBillingAndIAMPolicy",
```

```
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": {
                            "Effect": "Deny",
                            "Action": [
                                "aws-portal:*",
                                "iam:*"
                            ],
                            "Resource": "*"
                        }
                    }
                }
            ],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        },
        {
            "UserName": "Charlie",
            "GroupList": [
                "Dev"
            ],
            "CreateDate": "2013-10-14T18:33:56Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Charlie"
        }
    ],
    "Policies": [
        {
            "PolicyName": "create-update-delete-set-managed-policies",
            "CreateDate": "2015-02-06T19:58:34Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2015-02-06T19:58:34Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
```

```
                        "Statement": {
                            "Effect": "Allow",
                            "Action": [
                                "iam:CreatePolicy",
                                "iam:CreatePolicyVersion",
                                "iam:DeletePolicy",
                                "iam:DeletePolicyVersion",
                                "iam:GetPolicy",
                                "iam:GetPolicyVersion",
                                "iam:ListPolicies",
                                "iam:ListPolicyVersions",
                                "iam:SetDefaultPolicyVersion"
                            ],
                            "Resource": "*"
                        }
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
            "UpdateDate": "2015-02-06T19:58:34Z"
        },
        {
            "PolicyName": "S3-read-only-specific-bucket",
            "CreateDate": "2015-01-21T21:39:41Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2015-01-21T21:39:41Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Action": [
                                    "s3:Get*",
                                    "s3:List*"
                                ],
```

```
                            "Resource": [
                                "arn:aws:s3:::amzn-s3-demo-bucket",
                                "arn:aws:s3:::amzn-s3-demo-bucket/*"
                            ]
                        }
                    ]
                },
                "IsDefaultVersion": true
            }
        ],
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-bucket",
        "UpdateDate": "2015-01-21T23:39:41Z"
    },
    {
        "PolicyName": "AmazonEC2FullAccess",
        "CreateDate": "2015-02-06T18:40:15Z",
        "AttachmentCount": 1,
        "IsAttachable": true,
        "PolicyId": "ANPA1234567890EXAMPLE",
        "DefaultVersionId": "v1",
        "PolicyVersionList": [
            {
                "CreateDate": "2014-10-30T20:59:46Z",
                "VersionId": "v1",
                "Document": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Action": "ec2:*",
                            "Effect": "Allow",
                            "Resource": "*"
                        },
                        {
                            "Effect": "Allow",
                            "Action": "elasticloadbalancing:*",
                            "Resource": "*"
                        },
                        {
                            "Effect": "Allow",
                            "Action": "cloudwatch:*",
                            "Resource": "*"
                        },
                        {
```

```
                                "Effect": "Allow",
                                "Action": "autoscaling:*",
                                "Resource": "*"
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
            "UpdateDate": "2015-02-06T18:40:15Z"
        }
    ],
    "Marker": "EXAMPLEkakv9BCuUNFDtxWSyfzetYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
}
```

Pour plus d'informations, veuillez consulter Consignes pour les audits de sécuritéAWS dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAccountAuthorizationDetailsà la section Référence des AWS CLI commandes.

`get-account-password-policy`

L'exemple de code suivant montre comment utiliser`get-account-password-policy`.

AWS CLI

Pour afficher la politique de mot de passe du compte actuel

La commande `get-account-password-policy` suivante affiche les détails de la politique de mot de passe du compte actuel.

```
aws iam get-account-password-policy
```

Sortie :

```
{
    "PasswordPolicy": {
```

```
        "AllowUsersToChangePassword": false,
        "RequireLowercaseCharacters": false,
        "RequireUppercaseCharacters": false,
        "MinimumPasswordLength": 8,
        "RequireNumbers": true,
        "RequireSymbols": true
    }
}
```

Si aucune politique de mot de passe n'est définie pour le compte, la commande renvoie une erreur `NoSuchEntity`.

Pour plus d'informations, consultez la section Définition d'une politique de mot de passe du compte pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAccountPasswordPolicyà la section Référence des AWS CLI commandes.

`get-account-summary`

L'exemple de code suivant montre comment utiliser`get-account-summary`.

AWS CLI

Pour récupérer des informations sur l'utilisation des entités IAM et les quotas IAM dans le compte actuel

La commande `get-account-summary` suivante renvoie des informations sur l'utilisation actuelle des entités IAM et les quotas IAM actuels dans le compte.

```
aws iam get-account-summary
```

Sortie :

```
{
    "SummaryMap": {
        "UsersQuota": 5000,
        "GroupsQuota": 100,
        "InstanceProfiles": 6,
        "SigningCertificatesPerUserQuota": 2,
        "AccountAccessKeysPresent": 0,
```

```
        "RolesQuota": 250,
        "RolePolicySizeQuota": 10240,
        "AccountSigningCertificatesPresent": 0,
        "Users": 27,
        "ServerCertificatesQuota": 20,
        "ServerCertificates": 0,
        "AssumeRolePolicySizeQuota": 2048,
        "Groups": 7,
        "MFADevicesInUse": 1,
        "Roles": 3,
        "AccountMFAEnabled": 1,
        "MFADevices": 3,
        "GroupsPerUserQuota": 10,
        "GroupPolicySizeQuota": 5120,
        "InstanceProfilesQuota": 100,
        "AccessKeysPerUserQuota": 2,
        "Providers": 0,
        "UserPolicySizeQuota": 2048
    }
}
```

Pour plus d'informations sur les limites des entités, consultez les quotas IAM et AWS STS dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAccountSummaryà la section Référence des AWS CLI commandes.

`get-context-keys-for-custom-policy`

L'exemple de code suivant montre comment utiliser`get-context-keys-for-custom-policy`.

AWS CLI

Exemple 1 : pour répertorier les clés de contexte référencées par une ou plusieurs politiques JSON personnalisées fournies en paramètre sur la ligne de commande

La commande `get-context-keys-for-custom-policy` suivante analyse chaque politique fournie et répertorie les clés de contexte utilisées par ces politiques. Utilisez cette commande pour identifier les valeurs de clé de contexte que vous devez fournir pour utiliser correctement les commandes `simulate-custom-policy` et `simulate-custom-policy` du simulateur de politiques. Vous pouvez également récupérer la liste des clés de contexte utilisées par toutes les politiques associées à un utilisateur ou à un rôle IAM à l'aide de la commande `get-context-`

`keys-for-custom-policy`. Les valeurs des paramètres qui commencent par `file://` ordonnent à la commande de lire le fichier et d'utiliser son contenu comme valeur du paramètre au lieu du nom du fichier lui-même.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/${aws:username}","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
```

Sortie :

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Exemple 2 : pour répertorier les clés de contexte référencées par une ou plusieurs politiques JSON personnalisées fournies en entrée de fichier

La commande `get-context-keys-for-custom-policy` suivante est identique à l'exemple précédent, sauf que les politiques sont fournies dans un fichier plutôt que sous forme de paramètre. Comme la commande attend une liste de chaînes JSON, et non une liste de structures JSON, le fichier doit être structuré comme suit, bien que vous puissiez le réduire en une seule unité.

```
[
    "Policy1",
    "Policy2"
]
```

Ainsi, un fichier contenant la politique de l'exemple précédent doit ressembler à ce qui suit. Vous devez échapper chaque guillemet double intégré dans la chaîne de politique en le faisant précéder d'une barre oblique inverse ''.

```
[ "{\"Version\": \"2012-10-17\", \"Statement\": {\"Effect\": \"Allow\", \"Action
\": \"dynamodb:*\", \"Resource\": \"arn:aws:dynamodb:us-west-2:128716708097:table/
```

```
${aws:username}\", \"Condition\": {\"DateGreaterThan\": {\"aws:CurrentTime\":
 \"2015-08-16T12:00:00Z\"}}}}" ]
```

Ce fichier peut ensuite être soumis à la commande suivante.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list file://policyfile.json
```

Sortie :

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du simulateur de politique IAM (AWS CLI et AWS API) dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetContextKeysForCustomPolicyà la section Référence des AWS CLI commandes.

`get-context-keys-for-principal-policy`

L'exemple de code suivant montre comment utiliser`get-context-keys-for-principal-policy`.

AWS CLI

Pour répertorier les clés de contexte référencées par toutes les politiques associées à un principal IAM

La commande `get-context-keys-for-principal-policy` suivante permet de récupérer toutes les politiques associées à l'utilisateur `saanvi` et aux groupes dont elle est membre. Elle analyse ensuite chacune d'elles et répertorie les clés de contexte utilisées par ces politiques. Utilisez cette commande pour identifier les valeurs de clé de contexte que vous devez fournir pour utiliser correctement les commandes `simulate-custom-policy` et `simulate-principal-policy`. Vous pouvez également récupérer la liste des clés de contexte utilisées par une politique JSON arbitraire à l'aide de la commande `get-context-keys-for-custom-policy`.

```
aws iam get-context-keys-for-principal-policy \
   --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

Sortie :

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du simulateur de politique IAM (AWS CLI et AWS API) dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetContextKeysForPrincipalPolicyà la section Référence des AWS CLI commandes.

`get-credential-report`

L'exemple de code suivant montre comment utiliser`get-credential-report`.

AWS CLI

Pour obtenir un rapport sur les informations d’identification

Cet exemple ouvre le rapport renvoyé et le transmet au pipeline sous la forme d’un tableau de lignes de texte.

```
aws iam get-credential-report
```

Sortie :

```
{
    "GeneratedTime":  "2015-06-17T19:11:50Z",
    "ReportFormat": "text/csv"
}
```

Pour plus d'informations, consultez la section Obtenir des rapports d'identification pour votre AWS compte dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetCredentialReportà la section Référence des AWS CLI commandes.

`get-group-policy`

L'exemple de code suivant montre comment utiliser`get-group-policy`.

AWS CLI

Pour obtenir des informations sur une politique attachée à un groupe IAM

La commande `get-group-policy` suivante permet d'obtenir des informations sur la politique spécifiée attachée au groupe nommé `Test-Group`.

```
aws iam get-group-policy \
    --group-name Test-Group \
    --policy-name S3-ReadOnly-Policy
```

Sortie :

```
{
    "GroupName": "Test-Group",
    "PolicyDocument": {
        "Statement": [
            {
                "Action": [
                    "s3:Get*",
                    "s3:List*"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "PolicyName": "S3-ReadOnly-Policy"
}
```

Pour plus d'informations, consultez Gestion des politiques IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetGroupPolicyà la section Référence des AWS CLI commandes.

`get-group`

L'exemple de code suivant montre comment utiliser`get-group`.

AWS CLI

Pour obtenir un groupe IAM

Cet exemple renvoie des informations sur le groupe IAM `Admins`.

```
aws iam get-group \
    --group-name Admins
```

Sortie :

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-06-16T19:41:48Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    },
    "Users": []
}
```

Pour plus d'informations, consultez Identités IAM (utilisateurs, groupes et rôles) dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetGroupà la section Référence des AWS CLI commandes.

`get-instance-profile`

L'exemple de code suivant montre comment utiliser`get-instance-profile`.

AWS CLI

Pour obtenir des informations sur un profil d'instance

La commande `get-instance-profile` suivante permet d'obtenir des informations sur le profil d'instance nommé `ExampleInstanceProfile`.

```
aws iam get-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

Sortie :

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",
        "Roles": [
            {
                "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                "RoleId": "AIDGPMS9RO4H3FEXAMPLE",
                "CreateDate": "2013-01-09T06:33:26Z",
                "RoleName": "Test-Role",
                "Path": "/",
                "Arn": "arn:aws:iam::336924118301:role/Test-Role"
            }
        ],
        "CreateDate": "2013-06-12T23:52:02Z",
        "InstanceProfileName": "ExampleInstanceProfile",
        "Path": "/",
        "Arn": "arn:aws:iam::336924118301:instance-profile/ExampleInstanceProfile"
    }
}
```

Pour de plus amples informations, consultez Utilisation de profils d'instance dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetInstanceProfileà la section Référence des AWS CLI commandes.

`get-login-profile`

L'exemple de code suivant montre comment utiliser`get-login-profile`.

AWS CLI

Pour obtenir des informations sur le mot de passe d'un utilisateur IAM

La commande `get-login-profile` suivante permet d'obtenir des informations sur le mot de passe de l'utilisateur IAM nommé Bob.

```
aws iam get-login-profile \
    --user-name Bob
```

Sortie :

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2012-09-21T23:03:39Z"
    }
}
```

La commande `get-login-profile` peut être utilisée pour vérifier qu'un utilisateur IAM possède un mot de passe. La commande renvoie une erreur `NoSuchEntity` si aucun mot de passe n'est défini pour l'utilisateur.

Vous ne pouvez pas afficher un mot de passe à l'aide de cette commande. Si le mot de passe est perdu, vous pouvez le réinitialiser (`update-login-profile`) pour l'utilisateur. Vous pouvez également supprimer le profil de connexion (`delete-login-profile`) de l'utilisateur, puis en recréer un (`create-login-profile`).

Pour de plus amples informations, consultez Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetLoginProfileà la section Référence des AWS CLI commandes.

`get-mfa-device`

L'exemple de code suivant montre comment utiliser`get-mfa-device`.

AWS CLI

Pour récupérer des informations sur une clé de sécurité FIDO

L'exemple de `get-mfa-device` commande suivant permet de récupérer des informations sur la clé de sécurité FIDO spécifiée.

```
aws iam get-mfa-device \
```

```
    --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-
EXAMPLEBN5FHTECLFG7EXAMPLE
```

Sortie :

```
{
    "UserName": "alice",
    "SerialNumber": "arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-
EXAMPLEBN5FHTECLFG7EXAMPLE",
    "EnableDate": "2023-09-19T01:49:18+00:00",
    "Certifications": {
        "FIDO": "L1"
    }
}
```

Pour plus d'informations, consultez Utilisation de l'authentification multifactorielle (MFA) dans AWS dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetMfaDeviceà la section Référence des AWS CLI commandes.

`get-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`get-open-id-connect-provider`.

AWS CLI

Pour renvoyer des informations sur le fournisseur OpenID Connect spécifié

Cet exemple renvoie des informations sur le fournisseur OpenID Connect dont l'ARN est `arn:aws:iam::123456789012:oidc-provider/server.example.com`.

```
aws iam get-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Sortie :

```
{
    "Url": "server.example.com"
```

```
        "CreateDate": "2015-06-16T19:41:48Z",
        "ThumbprintList": [
        "12345abcdefghijk67890lmnopqrst987example"
        ],
        "ClientIDList": [
        "example-application-ID"
        ]
}
```

Pour plus d'informations, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`get-organizations-access-report`

L'exemple de code suivant montre comment utiliser`get-organizations-access-report`.

AWS CLI

Pour récupérer un rapport d'accès

L'`get-organizations-access-report`exemple suivant affiche un rapport d'accès généré précédemment pour une entité AWS Organizations. Pour générer un rapport, utilisez la commande `generate-organizations-access-report`.

```
aws iam get-organizations-access-report \
    --job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

Sortie :

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-09-30T06:53:36.187Z",
    "JobCompletionDate": "2019-09-30T06:53:37.547Z",
    "NumberOfServicesAccessible": 188,
    "NumberOfServicesNotAccessed": 171,
    "AccessDetails": [
        {
            "ServiceName": "Alexa for Business",
```

```
            "ServiceNamespace": "a4b",
            "TotalAuthenticatedEntities": 0
        },
        ...
}
```

Pour plus d'informations, voir Affiner les autorisations lors de AWS l'utilisation des dernières informations consultées dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetOrganizationsAccessReportà la section Référence des AWS CLI commandes.

`get-policy-version`

L'exemple de code suivant montre comment utiliser`get-policy-version`.

AWS CLI

Pour récupérer des informations sur la version spécifiée de la politique gérée spécifiée

Cet exemple renvoie le document de politique pour la version v2 de la politique dont l'ARN est `arn:aws:iam::123456789012:policy/MyManagedPolicy`.

```
aws iam get-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Sortie :

```
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "iam:*",
                    "Resource": "*"
                }
            ]
        },
```

```
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2023-04-11T00:22:54+00:00"
    }
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetPolicyVersionà la section Référence des AWS CLI commandes.

`get-policy`

L'exemple de code suivant montre comment utiliser`get-policy`.

AWS CLI

Pour récupérer des informations sur la politique gérée spécifiée

Cet exemple renvoie des détails sur la politique gérée dont l'ARN est `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Sortie :

```
{
    "Policy": {
        "PolicyName": "MySamplePolicy",
        "CreateDate": "2015-06-17T19:23;32Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "Z27SI6FQMGNQ2EXAMPLE1",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
        "UpdateDate": "2015-06-17T19:23:32Z"
    }
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetPolicyà la section Référence des AWS CLI commandes.

`get-role-policy`

L'exemple de code suivant montre comment utiliser`get-role-policy`.

AWS CLI

Pour obtenir des informations sur une politique attachée à un rôle IAM

La commande `get-role-policy` suivante permet d'obtenir des informations sur la politique spécifiée attachée au rôle nommé `Test-Role`.

```
aws iam get-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

Sortie :

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
      "Statement": [
          {
              "Action": [
                  "s3:ListBucket",
                  "s3:Put*",
                  "s3:Get*",
                  "s3:*MultipartUpload*"
              ],
              "Resource": "*",
              "Effect": "Allow",
              "Sid": "1"
          }
      ]
  }
  "PolicyName": "ExamplePolicy"
}
```

Pour plus d'informations, consultez Création de rôles IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetRolePolicyà la section Référence des AWS CLI commandes.

`get-role`

L'exemple de code suivant montre comment utiliser`get-role`.

AWS CLI

Pour obtenir des informations sur un rôle IAM

La commande `get-role` suivante permet d'obtenir des informations sur le rôle nommé `Test-Role`.

```
aws iam get-role \
    --role-name Test-Role
```

Sortie :

```
{
    "Role": {
        "Description": "Test Role",
        "AssumeRolePolicyDocument":"<URL-encoded-JSON>",
        "MaxSessionDuration": 3600,
        "RoleId": "AROA1234567890EXAMPLE",
        "CreateDate": "2019-11-13T16:45:56Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "RoleLastUsed": {
            "Region": "us-east-1",
            "LastUsedDate": "2019-11-13T17:14:00Z"
        },
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

La commande affiche la politique d'approbation attachée au rôle. Pour répertorier les politiques des autorisations attachées à un rôle, utilisez la commande `list-role-policies`.

Pour plus d'informations, consultez Création de rôles IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetRoleà la section Référence des AWS CLI commandes.

`get-saml-provider`

L'exemple de code suivant montre comment utiliser`get-saml-provider`.

AWS CLI

Pour récupérer le métadocument du fournisseur SAML

Cet exemple extrait les informations relatives au fournisseur SAML 2.0 dont l'ARM est le nom `arn:aws:iam::123456789012:saml-provider/SAMLADFS`. La réponse inclut le document de métadonnées que vous avez obtenu du fournisseur d'identité pour créer l'entité du fournisseur AWS SAML ainsi que les dates de création et d'expiration.

```
aws iam get-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Sortie :

```
{
    "SAMLMetadataDocument": "...SAMLMetadataDocument-XML...",
    "CreateDate": "2017-03-06T22:29:46+00:00",
    "ValidUntil": "2117-03-06T22:29:46.433000+00:00",
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Pour plus d'informations, consultez Création de fournisseurs d'identité SAML IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetSamlProviderà la section Référence des AWS CLI commandes.

`get-server-certificate`

L'exemple de code suivant montre comment utiliser`get-server-certificate`.

AWS CLI

Pour obtenir des informations sur un certificat de serveur associé à votre AWS compte

La `get-server-certificate` commande suivante permet de récupérer tous les détails relatifs au certificat de serveur spécifié dans votre AWS compte.

```
aws iam get-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

Sortie :

```
{
    "ServerCertificate": {
        "ServerCertificateMetadata": {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        "CertificateBody": "-----BEGIN CERTIFICATE-----
            MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
            VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
            b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
            BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
            MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
            VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
            b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
            YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
            21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
            rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
            Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
            nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
            FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
            NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
        "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCCAfICCQD6md
```

```
            7oRw0uXOjANBgkqhkiG9w0BAqQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgT
            AldBMRAwDgYDVQQHEwdTZWF0drGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAs
            TC0lBTSBDb25zb2xlMRIwEAYDVsQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQ
            jb20wHhcNMTEwNDI1MjA0NTIxWhtcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
            MCVVMxCzAJBgNVBAgTAldBMRAwDgsYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBb
            WF6b24xFDASBgNVBAsTC0lBTSBDb2d5zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMx
            HzAdBgkqhkiG9w0BCQEWEG5vb25lQGfFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
            BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
            k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
            ITxOUSQv7c7ugFFDzQGBzZswY6786m86gjpEIbb3OhjZnzcvQAaRHhdlQWIMm2nr
            AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
            KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJIlJ00zbhNYS5f6Guo
            EDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
            3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
    }
}
```

Pour répertorier les certificats de serveur disponibles dans votre AWS compte, utilisez la `list-server-certificates` commande.

Pour de plus amples informations, veuillez consulter Gestion des certificats de serveur dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetServerCertificateà la section Référence des AWS CLI commandes.

`get-service-last-accessed-details-with-entities`

L'exemple de code suivant montre comment utiliser`get-service-last-accessed-details-with-entities`.

AWS CLI

Pour récupérer un rapport d'accès aux services contenant les détails d'un service

L'exemple `get-service-last-accessed-details-with-entities` suivant extrait un rapport contenant des informations détaillées sur les utilisateurs IAM et les autres entités ayant accédé au service spécifié. Pour générer un rapport, utilisez la commande `generate-service-last-accessed-details`. Pour obtenir une liste des services auxquels on accède à l'aide d'espaces de noms, utilisez `get-service-last-accessed-details`.

```
aws iam get-service-last-accessed-details-with-entities \
```

```
    --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \
    --service-namespace lambda
```

Sortie :

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:55:41.756Z",
    "JobCompletionDate": "2019-10-01T03:55:42.533Z",
    "EntityDetailsList": [
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/admin",
                "Name": "admin",
                "Type": "USER",
                "Id": "AIDAIO2XMPLENQEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-30T23:02:00Z"
        },
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/developer",
                "Name": "developer",
                "Type": "USER",
                "Id": "AIDAIBEYXMPL2YEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-16T19:34:00Z"
        }
    ]
}
```

Pour plus d'informations, voir Affiner les autorisations lors de AWS l'utilisation des dernières informations consultées dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetServiceLastAccessedDetailsWithEntitiesà la section Référence des AWS CLI commandes.

`get-service-last-accessed-details`

L'exemple de code suivant montre comment utiliser`get-service-last-accessed-details`.

AWS CLI

Pour récupérer un rapport d'accès aux services

L'exemple `get-service-last-accessed-details` suivant récupère un rapport généré précédemment qui répertorie les services auxquels accèdent les entités IAM. Pour générer un rapport, utilisez la commande `generate-service-last-accessed-details`.

```
aws iam get-service-last-accessed-details \
    --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

Sortie :

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:50:35.929Z",
    "ServicesLastAccessed": [
        ...
        {
            "ServiceName": "AWS Lambda",
            "LastAuthenticated": "2019-09-30T23:02:00Z",
            "ServiceNamespace": "lambda",
            "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/admin",
            "TotalAuthenticatedEntities": 6
        },
    ]
}
```

Pour plus d'informations, voir Affiner les autorisations lors de AWS l'utilisation des dernières informations consultées dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetServiceLastAccessedDetailsà la section Référence des AWS CLI commandes.

`get-service-linked-role-deletion-status`

L'exemple de code suivant montre comment utiliser`get-service-linked-role-deletion-status`.

AWS CLI

Pour vérifier le statut d'une requête de suppression d'un rôle lié à un service

L'exemple `get-service-linked-role-deletion-status` suivant affiche le statut d'une requête précédente de suppression d'un rôle lié à un service. L'opération de suppression s'effectue de manière asynchrone. Lorsque vous effectuez la requête, vous obtenez une valeur `DeletionTaskId` que vous fournissez en tant que paramètre de cette commande.

```
aws iam get-service-linked-role-deletion-status \
    --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

Sortie :

```
{
"Status": "SUCCEEDED"
}
```

Pour plus d'informations, consultez Utilisation des rôles liés à un service dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetServiceLinkedRoleDeletionStatusà la section Référence des AWS CLI commandes.

`get-ssh-public-key`

L'exemple de code suivant montre comment utiliser`get-ssh-public-key`.

AWS CLI

Exemple 1 : Pour récupérer une clé publique SSH attachée à un utilisateur IAM sous forme codée SSH

La `get-ssh-public-key` commande suivante récupère la clé publique SSH spécifiée auprès de l'utilisateur IAM. `sofia` La sortie est encodée en SSH.

```
aws iam get-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE \
    --encoding SSH
```

Sortie :

```
{
    "SSHPublicKey": {
        "UserName": "sofia",
        "SSHPublicKeyId": "APKA123456789EXAMPLE",
        "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
        "SSHPublicKeyBody": "ssh-rsa <<long encoded SSH string>>",
        "Status": "Inactive",
        "UploadDate": "2019-04-18T17:04:49+00:00"
    }
}
```

Exemple 2 : Pour récupérer une clé publique SSH attachée à un utilisateur IAM sous forme codée PEM

La `get-ssh-public-key` commande suivante récupère la clé publique SSH spécifiée auprès de l'utilisateur IAM. `sofia` La sortie est encodée au format PEM.

```
aws iam get-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE \
    --encoding PEM
```

Sortie :

```
{
    "SSHPublicKey": {
        "UserName": "sofia",
        "SSHPublicKeyId": "APKA123456789EXAMPLE",
        "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
        "SSHPublicKeyBody": ""-----BEGIN PUBLIC KEY-----\n<<long encoded PEM
 string>>\n-----END PUBLIC KEY-----\n"",
        "Status": "Inactive",
        "UploadDate": "2019-04-18T17:04:49+00:00"
    }
}
```

Pour plus d'informations, consultez la section Utiliser les clés SSH et SSH avec CodeCommit dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetSshPublicKeyà la section Référence des AWS CLI commandes.

`get-user-policy`

L'exemple de code suivant montre comment utiliser`get-user-policy`.

AWS CLI

Pour répertorier les détails de la politique d'un utilisateur IAM

La commande `get-user-policy` suivante répertorie les détails de la politique spécifiée qui est attachée à l'utilisateur IAM nommé Bob.

```
aws iam get-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

Sortie :

```
{
    "UserName": "Bob",
    "PolicyName": "ExamplePolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "*",
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
}
```

Pour obtenir une liste des politiques d'un utilisateur IAM, utilisez la commande `list-user-policies`.

Pour de plus amples informations, veuillez consulter Policies and permissions in IAM (Stratégies et autorisations dans IAM) dans le AWS IAM Guide de l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous GetUserPolicyà la section Référence des AWS CLI commandes.

`get-user`

L'exemple de code suivant montre comment utiliser`get-user`.

AWS CLI

Pour obtenir des informations sur un utilisateur IAM

La commande `get-user` suivante permet d'obtenir des informations sur l'utilisateur IAM nommé `Paulo`.

```
aws iam get-user \
    --user-name Paulo
```

Sortie :

```
{
    "User": {
        "UserName": "Paulo",
        "Path": "/",
        "CreateDate": "2019-09-21T23:03:13Z",
        "UserId": "AIDA123456789EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Paulo"
    }
}
```

Pour plus d'informations, consultez la section Gestion des utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetUserà la section Référence des AWS CLI commandes.

`list-access-keys`

L'exemple de code suivant montre comment utiliser`list-access-keys`.

AWS CLI

Pour répertorier la clé d'accès IDs d'un utilisateur IAM

La `list-access-keys` commande suivante répertorie les clés d'accès IDs pour l'utilisateur IAM nomméBob.

```
aws iam list-access-keys \
    --user-name Bob
```

Sortie :

```
{
    "AccessKeyMetadata": [
        {
            "UserName": "Bob",
            "Status": "Active",
            "CreateDate": "2013-06-04T18:17:34Z",
            "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CreateDate": "2013-06-06T20:42:26Z",
            "AccessKeyId": "AKIAI44QH8DHBEXAMPLE"
        }
    ]
}
```

Vous ne pouvez pas répertorier les clés d'accès secrètes des utilisateurs IAM. Si les clés d'accès secrètes sont perdues, vous devez créer de nouvelles clés d'accès à l'aide de la commande `create-access-keys`.

Pour de plus amples informations, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAccessKeysà la section Référence des AWS CLI commandes.

`list-account-aliases`

L'exemple de code suivant montre comment utiliser`list-account-aliases`.

AWS CLI

Pour répertorier des alias de compte

La commande `list-account-aliases` suivante répertorie les alias du compte actuel.

```
aws iam list-account-aliases
```

Sortie :

```
{
    "AccountAliases": [
    "mycompany"
    ]
}
```

Pour plus d'informations, consultez la section Votre identifiant de AWS compte et son alias dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAccountAliasesà la section Référence des AWS CLI commandes.

`list-attached-group-policies`

L'exemple de code suivant montre comment utiliser`list-attached-group-policies`.

AWS CLI

Pour répertorier toutes les politiques gérées attachées au groupe spécifié

Cet exemple renvoie les noms et ARNs les politiques gérées attachés au groupe IAM nommé `Admins` dans le AWS compte.

```
aws iam list-attached-group-policies \
    --group-name Admins
```

Sortie :

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
```

```
        }
    ],
    "IsTruncated": false
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAttachedGroupPoliciesà la section Référence des AWS CLI commandes.

`list-attached-role-policies`

L'exemple de code suivant montre comment utiliser`list-attached-role-policies`.

AWS CLI

Pour répertorier toutes les politiques gérées qui sont attachées au rôle spécifié

Cette commande renvoie les noms et ARNs les politiques gérées associées au rôle IAM nommé `SecurityAuditRole` dans le AWS compte.

```
aws iam list-attached-role-policies \
    --role-name SecurityAuditRole
```

Sortie :

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAttachedRolePoliciesà la section Référence des AWS CLI commandes.

`list-attached-user-policies`

L'exemple de code suivant montre comment utiliser`list-attached-user-policies`.

AWS CLI

Pour répertorier toutes les politiques gérées attachées à l'utilisateur spécifié

Cette commande renvoie les noms et ARNs les politiques gérées pour l'utilisateur IAM nommé Bob dans le AWS compte.

```
aws iam list-attached-user-policies \
    --user-name Bob
```

Sortie :

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAttachedUserPoliciesà la section Référence des AWS CLI commandes.

`list-entities-for-policy`

L'exemple de code suivant montre comment utiliser`list-entities-for-policy`.

AWS CLI

Pour répertorier tous les utilisateurs, groupes et rôles auxquels la politique gérée spécifiée est attachée

Cet exemple renvoie une liste des groupes, des rôles et des utilisateurs IAM auxquels la politique `arn:aws:iam::123456789012:policy/TestPolicy` est attachée.

```
aws iam list-entities-for-policy \
    --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

Sortie :

```
{
    "PolicyGroups": [
        {
            "GroupName": "Admins",
            "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyUsers": [
        {
            "UserName": "Alice",
            "UserId": "AIDACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyRoles": [
        {
            "RoleName": "DevRole",
            "RoleId": "AROADBQP57FF2AEXAMPLE"
        }
    ],
    "IsTruncated": false
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListEntitiesForPolicyà la section Référence des AWS CLI commandes.

`list-group-policies`

L'exemple de code suivant montre comment utiliser`list-group-policies`.

AWS CLI

Pour répertorier toutes les politiques en ligne attachées au groupe spécifié

La commande `list-group-policies` suivante répertorie les noms des politiques en ligne attachées au groupe IAM nommé `Admins` dans le compte actuel.

```
aws iam list-group-policies \
    --group-name Admins
```

Sortie :

```
{
    "PolicyNames": [
        "AdminRoot",
        "ExamplePolicy"
    ]
}
```

Pour plus d'informations, consultez Gestion des politiques IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListGroupPoliciesà la section Référence des AWS CLI commandes.

`list-groups-for-user`

L'exemple de code suivant montre comment utiliser`list-groups-for-user`.

AWS CLI

Pour répertorier les groupes auxquels l'utilisateur IAM appartient

La commande `list-groups-for-user` suivante affiche les groupes auxquels l'utilisateur IAM nommé Bob appartient.

```
aws iam list-groups-for-user \
    --user-name Bob
```

Sortie :

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:18:08Z",
            "GroupId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admin",
            "GroupName": "Admin"
        },
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:37:28Z",
            "GroupId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/s3-Users",
            "GroupName": "s3-Users"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des groupes IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListGroupsForUserà la section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour répertorier les groupes IAM du compte actuel

La commande `list-groups` suivante répertorie les groupes IAM du compte actuel.

```
aws iam list-groups
```

Sortie :

```
{
```

```
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-06-04T20:27:27.972Z",
            "GroupId": "AIDACKCEVSQ6C2EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "GroupName": "Admins"
        },
        {
            "Path": "/",
            "CreateDate": "2013-04-16T20:30:42Z",
            "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
            "GroupName": "S3-Admins"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des groupes IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListGroupsà la section Référence des AWS CLI commandes.

`list-instance-profile-tags`

L'exemple de code suivant montre comment utiliser`list-instance-profile-tags`.

AWS CLI

Pour répertorier les balises associées à un profil d'instance

La `list-instance-profile-tags` commande suivante permet de récupérer la liste des balises associées au profil d'instance spécifié.

```
aws iam list-instance-profile-tags \
    --instance-profile-name deployment-role
```

Sortie :

```
{
    "Tags": [
        {
```

```
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListInstanceProfileTagsà la section Référence des AWS CLI commandes.

`list-instance-profiles-for-role`

L'exemple de code suivant montre comment utiliser`list-instance-profiles-for-role`.

AWS CLI

Pour répertorier les profils d'instance d'un rôle IAM

La commande `list-instance-profiles-for-role` suivante répertorie les profils d'instance associés au rôle `Test-Role`.

```
aws iam list-instance-profiles-for-role \
    --role-name Test-Role
```

Sortie :

```
{
    "InstanceProfiles": [
        {
            "InstanceProfileId": "AIDGPMS9RO4H3FEXAMPLE",
            "Roles": [
                {
                    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                    "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
                    "CreateDate": "2013-06-07T20:42:15Z",
                    "RoleName": "Test-Role",
                    "Path": "/",
```

```
                    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
                }
            ],
            "CreateDate": "2013-06-07T21:05:24Z",
            "InstanceProfileName": "ExampleInstanceProfile",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:instance-profile/
ExampleInstanceProfile"
        }
    ]
}
```

Pour de plus amples informations, consultez Utilisation de profils d'instance dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListInstanceProfilesForRoleà la section Référence des AWS CLI commandes.

`list-instance-profiles`

L'exemple de code suivant montre comment utiliser`list-instance-profiles`.

AWS CLI

Pour répertorier les profils d'instance du compte

La commande `list-instance-profiles` suivante répertorie les profils d'instance associés au compte actuel.

```
aws iam list-instance-profiles
```

Sortie :

```
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "example-dev-role",
            "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-dev-role",
            "CreateDate": "2023-09-21T18:17:41+00:00",
            "Roles": [
                {
```

```
                "Path": "/",
                "RoleName": "example-dev-role",
                "RoleId": "AROAJ52OTH4H7LEXAMPLE",
                "Arn": "arn:aws:iam::123456789012:role/example-dev-role",
                "CreateDate": "2023-09-21T18:17:40+00:00",
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": "ec2.amazonaws.com"
                            },
                            "Action": "sts:AssumeRole"
                        }
                    ]
                }
            }
        ]
    },
    {
        "Path": "/",
        "InstanceProfileName": "example-s3-role",
        "InstanceProfileId": "AIPAJVJVNRIQFREXAMPLE",
        "Arn": "arn:aws:iam::123456789012:instance-profile/example-s3-role",
        "CreateDate": "2023-09-21T18:18:50+00:00",
        "Roles": [
            {
                "Path": "/",
                "RoleName": "example-s3-role",
                "RoleId": "AROAINUBC5O7XLEXAMPLE",
                "Arn": "arn:aws:iam::123456789012:role/example-s3-role",
                "CreateDate": "2023-09-21T18:18:49+00:00",
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": "ec2.amazonaws.com"
                            },
                            "Action": "sts:AssumeRole"
                        }
                    ]
```

```
                    }
                }
            ]
        }
    ]
}
```

Pour de plus amples informations, consultez Utilisation de profils d'instance dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListInstanceProfilesà la section Référence des AWS CLI commandes.

`list-mfa-device-tags`

L'exemple de code suivant montre comment utiliser`list-mfa-device-tags`.

AWS CLI

Pour répertorier les tags attachés à un périphérique MFA

La `list-mfa-device-tags` commande suivante permet de récupérer la liste des balises associées au périphérique MFA spécifié.

```
aws iam list-mfa-device-tags \
    --serial-number arn:aws:iam::123456789012:mfa/alice
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListMfaDeviceTagsà la section Référence des AWS CLI commandes.

`list-mfa-devices`

L'exemple de code suivant montre comment utiliser`list-mfa-devices`.

AWS CLI

Pour répertorier tous les dispositifs MFA d'un utilisateur spécifique

Cet exemple renvoie des informations sur le dispositif MFA attribué à l'utilisateur IAM Bob.

```
aws iam list-mfa-devices \
    --user-name Bob
```

Sortie :

```
{
    "MFADevices": [
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Bob",
            "EnableDate": "2019-10-28T20:37:09+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "GAKT12345678",
            "EnableDate": "2023-02-18T21:44:42+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey1-7XNL7NFNLZ123456789EXAMPLE",
            "EnableDate": "2023-09-19T02:25:35+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey2-VDRQTDBBN5123456789EXAMPLE",
```

```
            "EnableDate": "2023-09-19T01:49:18+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez Utilisation de l'authentification multifactorielle (MFA) dans AWS dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListMfaDevicesà la section Référence des AWS CLI commandes.

`list-open-id-connect-provider-tags`

L'exemple de code suivant montre comment utiliser`list-open-id-connect-provider-tags`.

AWS CLI

Pour répertorier les tags attachés à un fournisseur d'identité compatible OpenID Connect (OIDC)

La `list-open-id-connect-provider-tags` commande suivante permet de récupérer la liste des balises associées au fournisseur d'identité OIDC spécifié.

```
aws iam list-open-id-connect-provider-tags \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListOpenIdConnectProviderTagsà la section Référence des AWS CLI commandes.

`list-open-id-connect-providers`

L'exemple de code suivant montre comment utiliser`list-open-id-connect-providers`.

AWS CLI

Pour répertorier les informations relatives aux fournisseurs OpenID Connect dans le compte AWS

Cet exemple renvoie une liste des ARNS de tous les fournisseurs OpenID Connect définis dans AWS le compte courant.

```
aws iam list-open-id-connect-providers
```

Sortie :

```
{
    "OpenIDConnectProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
        }
    ]
}
```

Pour plus d'informations, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListOpenIdConnectProvidersà la section Référence des AWS CLI commandes.

`list-organizations-features`

L'exemple de code suivant montre comment utiliser`list-organizations-features`.

AWS CLI

Pour répertorier les fonctionnalités d'accès root centralisé activées pour votre organisation

La `list-organizations-features` commande suivante répertorie les fonctionnalités d'accès root centralisé activées pour votre organisation.

```
aws iam list-organizations-features
```

Sortie :

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement",
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

Pour plus d'informations, consultez la section Gestion centralisée de l'accès root pour les comptes des membres dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListOrganizationsFeaturesà la section Référence des AWS CLI commandes.

`list-policies-granting-service-access`

L'exemple de code suivant montre comment utiliser`list-policies-granting-service-access`.

AWS CLI

Pour répertorier les politiques qui accordent un accès principal au service spécifié

L'`list-policies-granting-service-access`exemple suivant extrait la liste des politiques qui accordent à l'utilisateur IAM l'`sofia`accès au AWS CodeCommit service.

```
aws iam list-policies-granting-service-access \
    --arn arn:aws:iam::123456789012:user/sofia \
    --service-namespaces codecommit
```

Sortie :

```
{
    "PoliciesGrantingServiceAccess": [
        {
            "ServiceNamespace": "codecommit",
            "Policies": [
                {
                    "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
                    "PolicyType": "INLINE",
                    "EntityType": "USER",
                    "EntityName": "sofia"
                }
            ]
        }
    ],
    "IsTruncated": false
}
```

Pour plus d'informations, consultez la section Utilisation d'IAM avec CodeCommit : informations d'identification Git, clés SSH et clés AWS d'accès dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListPoliciesGrantingServiceAccessà la section Référence des AWS CLI commandes.

`list-policies`

L'exemple de code suivant montre comment utiliser`list-policies`.

AWS CLI

Pour répertorier les politiques gérées disponibles pour votre AWS compte

Cet exemple renvoie une collection des deux premières politiques gérées disponibles dans le AWS compte courant.

```
aws iam list-policies \
    --max-items 3
```

Sortie :

```
{
    "Policies": [
        {
```

```
            "PolicyName": "AWSCloudTrailAccessPolicy",
            "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 0,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2019-09-04T17:43:42+00:00",
            "UpdateDate": "2019-09-04T17:43:42+00:00"
        },
        {
            "PolicyName": "AdministratorAccess",
            "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 6,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:46+00:00",
            "UpdateDate": "2015-02-06T18:39:46+00:00"
        },
        {
            "PolicyName": "PowerUserAccess",
            "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
            "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
            "Path": "/",
            "DefaultVersionId": "v5",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:47+00:00",
            "UpdateDate": "2023-07-06T22:04:00+00:00"
        }
    ],
    "NextToken": "EXAMPLErZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiA4fQ=="
}
```


Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListPoliciesà la section Référence des AWS CLI commandes.

`list-policy-tags`

L'exemple de code suivant montre comment utiliser`list-policy-tags`.

AWS CLI

Pour répertorier les balises associées à une politique gérée

La `list-policy-tags` commande suivante permet de récupérer la liste des balises associées à la politique gérée spécifiée.

```
aws iam list-policy-tags \
    --policy-arn arn:aws:iam::123456789012:policy/billing-access
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListPolicyTagsà la section Référence des AWS CLI commandes.

`list-policy-versions`

L'exemple de code suivant montre comment utiliser`list-policy-versions`.

AWS CLI

Pour récupérer des informations sur les versions de la politique gérée spécifiée

Cet exemple renvoie la liste des versions disponibles de la politique dont l'ARN est `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam list-policy-versions \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Sortie :

```
{
    "IsTruncated": false,
    "Versions": [
        {
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2015-06-02T23:19:44Z"
        },
        {
        "VersionId": "v1",
        "IsDefaultVersion": false,
        "CreateDate": "2015-06-02T22:30:47Z"
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter Politiques et autorisations dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListPolicyVersionsà la section Référence des AWS CLI commandes.

`list-role-policies`

L'exemple de code suivant montre comment utiliser`list-role-policies`.

AWS CLI

Pour répertorier les politiques attachées à un rôle IAM

La commande `list-role-policies` suivante répertorie les noms des politiques des autorisations pour le rôle IAM spécifié.

```
aws iam list-role-policies \
```

```
    --role-name Test-Role
```

Sortie :

```
{
    "PolicyNames": [
        "ExamplePolicy"
    ]
}
```

Pour voir la politique d'approbation attachée à un rôle, utilisez la commande `get-role`. Pour voir les détails d'une politique des autorisations, utilisez la commande `get-role-policy`.

Pour plus d'informations, consultez Création de rôles IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListRolePoliciesà la section Référence des AWS CLI commandes.

`list-role-tags`

L'exemple de code suivant montre comment utiliser`list-role-tags`.

AWS CLI

Pour répertorier les balises attachées à un rôle

La commande `list-role-tags` suivante permet de récupérer la liste des balises associées au rôle spécifié.

```
aws iam list-role-tags \
    --role-name production-role
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
```

```
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListRoleTagsà la section Référence des AWS CLI commandes.

`list-roles`

L'exemple de code suivant montre comment utiliser`list-roles`.

AWS CLI

Pour répertorier les rôles IAM du compte actuel

La commande `list-roles` suivante répertorie les rôles IAM du compte actuel.

```
aws iam list-roles
```

Sortie :

```
{
    "Roles": [
        {
            "Path": "/",
            "RoleName": "ExampleRole",
            "RoleId": "AROAJ52OTH4H7LEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
            "CreateDate": "2017-09-12T19:23:36+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
```

```
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/example_path/",
            "RoleName": "ExampleRoleWithPath",
            "RoleId": "AROAI4QRP7UFT7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/example_path/
ExampleRoleWithPath",
            "CreateDate": "2023-09-21T20:29:38+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```

Pour plus d'informations, consultez Création de rôles IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListRolesà la section Référence des AWS CLI commandes.

`list-saml-provider-tags`

L'exemple de code suivant montre comment utiliser`list-saml-provider-tags`.

AWS CLI

Pour répertorier les balises associées à un fournisseur SAML

La `list-saml-provider-tags` commande suivante permet de récupérer la liste des balises associées au fournisseur SAML spécifié.

```
aws iam list-saml-provider-tags \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListSamlProviderTagsà la section Référence des AWS CLI commandes.

`list-saml-providers`

L'exemple de code suivant montre comment utiliser`list-saml-providers`.

AWS CLI

Pour répertorier les fournisseurs SAML dans le compte AWS

Cet exemple extrait la liste des fournisseurs SAML 2.0 créés dans le compte courant AWS .

```
aws iam list-saml-providers
```

Sortie :

```
{
    "SAMLProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
            "ValidUntil": "2015-06-05T22:45:14Z",
            "CreateDate": "2015-06-05T22:45:14Z"
        }
    ]
}
```

Pour plus d'informations, consultez Création de fournisseurs d'identité SAML IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, consultez la section Liste SAMLProviders dans AWS CLI la référence des commandes.

`list-server-certificate-tags`

L'exemple de code suivant montre comment utiliser`list-server-certificate-tags`.

AWS CLI

Pour répertorier les balises associées à un certificat de serveur

La `list-server-certificate-tags` commande suivante permet de récupérer la liste des balises associées au certificat de serveur spécifié.

```
aws iam list-server-certificate-tags \
    --server-certificate-name ExampleCertificate
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
```

```
    ]
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListServerCertificateTagsà la section Référence des AWS CLI commandes.

`list-server-certificates`

L'exemple de code suivant montre comment utiliser`list-server-certificates`.

AWS CLI

Pour répertorier les certificats de serveur de votre AWS compte

La `list-server-certificates` commande suivante répertorie tous les certificats de serveur stockés et disponibles pour utilisation dans votre AWS compte.

```
aws iam list-server-certificates
```

Sortie :

```
{
    "ServerCertificateMetadataList": [
        {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        {
            "Path": "/cloudfront/",
            "ServerCertificateName": "MyTestCert",
            "ServerCertificateId": "ASCAEXAMPLE456EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyTestCert",
            "UploadDate": "2015-04-21T18:14:16+00:00",
```

```
            "Expiration": "2018-01-14T17:52:36+00:00"
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter Gestion des certificats de serveur dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListServerCertificatesà la section Référence des AWS CLI commandes.

`list-service-specific-credential`

L'exemple de code suivant montre comment utiliser`list-service-specific-credential`.

AWS CLI

Exemple 1 : Répertorier les informations d'identification spécifiques au service pour un utilisateur

L'`list-service-specific-credentials`exemple suivant affiche toutes les informations d'identification spécifiques au service attribuées à l'utilisateur spécifié. Les mots de passe ne sont pas inclus dans la réponse.

```
aws iam list-service-specific-credentials \
    --user-name sofia
```

Sortie :

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

Exemple 2 : Répertorier les informations d'identification spécifiques au service pour un utilisateur filtré vers un service spécifique

L'`list-service-specific-credentials`exemple suivant affiche les informations d'identification spécifiques au service attribuées à l'utilisateur qui fait la demande. La liste est filtrée pour inclure uniquement les informations d'identification pour le service spécifié. Les mots de passe ne sont pas inclus dans la réponse.

```
aws iam list-service-specific-credentials \
    --service-name codecommit.amazonaws.com
```

Sortie :

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

Pour plus d'informations, consultez la section Créer des informations d'identification Git pour les connexions HTTPS CodeCommit dans le Guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListServiceSpecificCredentialà la section Référence des AWS CLI commandes.

`list-service-specific-credentials`

L'exemple de code suivant montre comment utiliser`list-service-specific-credentials`.

AWS CLI

Pour récupérer une liste d'informations d'identification

L'`list-service-specific-credentials`exemple suivant répertorie les informations d'identification générées pour l'accès HTTPS aux AWS CodeCommit référentiels pour un utilisateur nommé`developer`.

```
aws iam list-service-specific-credentials \
    --user-name developer \
```

```
    --service-name codecommit.amazonaws.com
```

Sortie :

```
{
    "ServiceSpecificCredentials": [
        {
            "UserName": "developer",
            "Status": "Inactive",
            "ServiceUserName": "developer-at-123456789012",
            "CreateDate": "2019-10-01T04:31:41Z",
            "ServiceSpecificCredentialId": "ACCAQFODXMPL4YFHP7DZE",
            "ServiceName": "codecommit.amazonaws.com"
        },
        {
            "UserName": "developer",
            "Status": "Active",
            "ServiceUserName": "developer+1-at-123456789012",
            "CreateDate": "2019-10-01T04:31:45Z",
            "ServiceSpecificCredentialId": "ACCAQFOXMPL6VW57M7AJP",
            "ServiceName": "codecommit.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, consultez la section Créer des informations d'identification Git pour les connexions HTTPS CodeCommit dans le Guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListServiceSpecificCredentialsà la section Référence des AWS CLI commandes.

`list-signing-certificates`

L'exemple de code suivant montre comment utiliser`list-signing-certificates`.

AWS CLI

Pour répertorier les certificats de signature d'un utilisateur IAM

La commande `list-signing-certificates` suivante répertorie les certificats de signature de l'utilisateur IAM nommé Bob.

```
aws iam list-signing-certificates \
```

```
    --user-name Bob
```

Sortie :

```
{
    "Certificates": [
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----
END CERTIFICATE-----",
            "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
            "UploadDate": "2013-06-06T21:40:08Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les certificats de signature dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListSigningCertificatesà la section Référence des AWS CLI commandes.

`list-ssh-public-keys`

L'exemple de code suivant montre comment utiliser`list-ssh-public-keys`.

AWS CLI

Pour répertorier les clés publiques SSH associées à un utilisateur IAM

L'`list-ssh-public-keys`exemple suivant répertorie les clés publiques SSH associées à l'utilisateur IAM. `sofia`

```
aws iam list-ssh-public-keys \
    --user-name sofia
```

Sortie :

```
{
    "SSHPublicKeys": [
        {
```

```
            "UserName": "sofia",
            "SSHPublicKeyId": "APKA1234567890EXAMPLE",
            "Status": "Inactive",
            "UploadDate": "2019-04-18T17:04:49+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utiliser les clés SSH et SSH avec CodeCommit dans le guide de l'utilisateur AWS IAM

- Pour plus de détails sur l'API, reportez-vous ListSshPublicKeysà la section Référence des AWS CLI commandes.

`list-user-policies`

L'exemple de code suivant montre comment utiliser`list-user-policies`.

AWS CLI

Pour répertorier les politiques d'un utilisateur IAM

La commande `list-user-policies` suivante répertorie les politiques attachées à l'utilisateur IAM nommé Bob.

```
aws iam list-user-policies \
    --user-name Bob
```

Sortie :

```
{
    "PolicyNames": [
        "ExamplePolicy",
        "TestPolicy"
    ]
}
```

Pour plus d'informations, consultez la section Création d'un utilisateur IAM dans votre AWS compte dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListUserPoliciesà la section Référence des AWS CLI commandes.

`list-user-tags`

L'exemple de code suivant montre comment utiliser`list-user-tags`.

AWS CLI

Pour répertorier les balises attachées à un utilisateur

La commande `list-user-tags` suivante extrait la liste des balises associées à l'utilisateur IAM spécifié.

```
aws iam list-user-tags \
    --user-name alice
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListUserTagsà la section Référence des AWS CLI commandes.

`list-users`

L'exemple de code suivant montre comment utiliser`list-users`.

AWS CLI

Pour répertorier les utilisateurs IAM

La commande `list-users` suivante répertorie les utilisateurs IAM du compte actuel.

```
aws iam list-users
```

Sortie :

```
{
    "Users": [
        {
            "UserName": "Adele",
            "Path": "/",
            "CreateDate": "2013-03-07T05:14:48Z",
            "UserId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Adele"
        },
        {
            "UserName": "Bob",
            "Path": "/",
            "CreateDate": "2012-09-21T23:03:13Z",
            "UserId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        }
    ]
}
```

Pour plus d'informations, consultez la section Liste des utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`list-virtual-mfa-devices`

L'exemple de code suivant montre comment utiliser`list-virtual-mfa-devices`.

AWS CLI

Pour afficher la liste des dispositifs MFA virtuels

La commande `list-virtual-mfa-devices` suivante répertorie les dispositifs MFA virtuels qui ont été configurés pour le compte actuel.

```
aws iam list-virtual-mfa-devices
```

Sortie :

```
{
    "VirtualMFADevices": [
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
        },
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
        }
    ]
}
```

Pour de plus d'informations, consultez Activation d'un dispositif virtuel d'authentification multifactorielle (MFA) dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListVirtualMfaDevicesà la section Référence des AWS CLI commandes.

`put-group-policy`

L'exemple de code suivant montre comment utiliser`put-group-policy`.

AWS CLI

Pour ajouter une politique à un groupe

La commande `put-group-policy` suivante ajoute une politique au groupe IAM nommé `Admins`.

```
aws iam put-group-policy \
    --group-name Admins \
    --policy-document file://AdminPolicy.json \
    --policy-name AdminRoot
```

Cette commande ne produit aucun résultat.

La politique est définie sous la forme d'un document JSON dans le fichier AdminPolicy.json. (Le nom et l'extension du fichier n'ont aucune importance.)

Pour plus d'informations, consultez Gestion des politiques IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous PutGroupPolicyà la section Référence des AWS CLI commandes.

`put-role-permissions-boundary`

L'exemple de code suivant montre comment utiliser`put-role-permissions-boundary`.

AWS CLI

Exemple 1 : pour appliquer une limite des autorisations basée sur une politique personnalisée à un rôle IAM

L'exemple `put-role-permissions-boundary` suivant applique la politique personnalisée nommée `intern-boundary` comme limite des autorisations pour le rôle IAM spécifié.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --role-name lambda-application-role
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour appliquer une limite d'autorisation basée sur une politique AWS gérée à un rôle IAM

L'`put-role-permissions-boundary`exemple suivant applique la `PowerUserAccess` politique AWS gérée comme limite d'autorisations pour le rôle IAM spécifié.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --role-name x-account-admin
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modification d'un rôle dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous PutRolePermissionsBoundaryà la section Référence des AWS CLI commandes.

`put-role-policy`

L'exemple de code suivant montre comment utiliser`put-role-policy`.

AWS CLI

Pour attacher une politique d’autorisation à un rôle IAM

La commande `put-role-policy` suivante ajoute une politique d’autorisation au rôle nommé `Test-Role`.

```
aws iam put-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

Cette commande ne produit aucun résultat.

La politique est définie sous la forme d'un document JSON dans le fichier AdminPolicy.json. (Le nom et l’extension du fichier n’ont aucune importance.)

Pour attacher une politique d’approbation à un rôle, utilisez la commande `update-assume-role-policy`.

Pour plus d’informations, consultez Modification d’un rôle dans le Guide de l’utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous PutRolePolicyà la section Référence des AWS CLI commandes.

`put-user-permissions-boundary`

L'exemple de code suivant montre comment utiliser`put-user-permissions-boundary`.

AWS CLI

Exemple 1 : pour appliquer une limite des autorisations basée sur une politique personnalisée à un utilisateur IAM

L’exemple `put-user-permissions-boundary` suivant applique une politique personnalisée nommée `intern-boundary` comme limite des autorisations pour l’utilisateur IAM spécifié.

```
aws iam put-user-permissions-boundary \
```

```
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --user-name intern
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour appliquer une limite d'autorisation basée sur une politique AWS gérée à un utilisateur IAM

L'`put-user-permissions-boundary`exemple suivant applique la politique AWS gérée `PowerUserAccess` nommée limite d'autorisations pour l'utilisateur IAM spécifié.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --user-name developer
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la rubrique Ajout et suppression d'autorisations basées sur l'identité IAM du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous PutUserPermissionsBoundaryà la section Référence des AWS CLI commandes.

`put-user-policy`

L'exemple de code suivant montre comment utiliser`put-user-policy`.

AWS CLI

Pour attacher une politique à un utilisateur IAM

La commande `put-user-policy` suivante attache une politique à l'utilisateur IAM nommé `Bob`.

```
aws iam put-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

Cette commande ne produit aucun résultat.

La politique est définie sous la forme d'un document JSON dans le fichier AdminPolicy.json. (Le nom et l'extension du fichier n'ont aucune importance.)

Pour plus d'informations, consultez la rubrique Ajout et suppression d'autorisations basées sur l'identité IAM du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous PutUserPolicyà la section Référence des AWS CLI commandes.

`remove-client-id-from-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`remove-client-id-from-open-id-connect-provider`.

AWS CLI

Pour supprimer l'ID client spécifié de la liste des clients IDs enregistrés pour le fournisseur IAM OpenID Connect spécifié

Cet exemple supprime l'ID client `My-TestApp-3` de la liste des clients IDs associés au fournisseur IAM OIDC dont l'ARN est l'ARN. `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`

```
aws iam remove-client-id-from-open-id-connect-provider
    --client-id My-TestApp-3 \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous RemoveClientIdFromOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`remove-role-from-instance-profile`

L'exemple de code suivant montre comment utiliser`remove-role-from-instance-profile`.

AWS CLI

Pour supprimer un rôle d'un profil d'instance

La commande `remove-role-from-instance-profile` suivante supprime la politique nommée `Test-Role` du profil d'instance nommé `ExampleInstanceProfile`.

```
aws iam remove-role-from-instance-profile \
    --instance-profile-name ExampleInstanceProfile \
    --role-name Test-Role
```

Pour de plus amples informations, consultez Utilisation de profils d'instance dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous RemoveRoleFromInstanceProfileà la section Référence des AWS CLI commandes.

`remove-user-from-group`

L'exemple de code suivant montre comment utiliser`remove-user-from-group`.

AWS CLI

Pour supprimer un utilisateur d'un groupe IAM

La commande `remove-user-from-group` suivante supprime l'utilisateur nommé `Bob` du groupe IAM nommé `Admins`.

```
aws iam remove-user-from-group \
    --user-name Bob \
    --group-name Admins
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Ajout et suppression d'utilisateurs dans un groupe IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous RemoveUserFromGroupà la section Référence des AWS CLI commandes.

`reset-service-specific-credential`

L'exemple de code suivant montre comment utiliser`reset-service-specific-credential`.

AWS CLI

Exemple 1 : réinitialisation du mot de passe pour un identifiant spécifique au service attaché à l'utilisateur qui fait la demande

L'`reset-service-specific-credential`exemple suivant génère un nouveau mot de passe cryptographiquement fort pour les informations d'identification spécifiques au service spécifiées associées à l'utilisateur qui fait la demande.

```
aws iam reset-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Sortie :

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "+oaFsNk7tLco+C/obP9GhhcOzGcKOayTmE3LnAmAmH4=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

Exemple 2 : réinitialiser le mot de passe d'un identifiant spécifique à un service associé à un utilisateur spécifié

L'`reset-service-specific-credential`exemple suivant génère un nouveau mot de passe cryptographiquement fort pour un identifiant spécifique au service attaché à l'utilisateur spécifié.

```
aws iam reset-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Sortie :

```
{
    "ServiceSpecificCredential": {
```

```
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "+oaFsNk7tLco+C/obP9GhhcOzGcKOayTmE3LnAmAmH4=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

Pour plus d'informations, consultez la section Créer des informations d'identification Git pour les connexions HTTPS CodeCommit dans le Guide de AWS CodeCommit l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous ResetServiceSpecificCredentialà la section Référence des AWS CLI commandes.

`resync-mfa-device`

L'exemple de code suivant montre comment utiliser`resync-mfa-device`.

AWS CLI

Pour resynchroniser un dispositif MFA

L'exemple `resync-mfa-device` suivant synchronise le dispositif MFA associé à l'utilisateur IAM Bob, et dont l'ARN est `arn:aws:iam::123456789012:mfa/BobsMFADevice`, avec un programme d'authentification qui a fourni les deux codes d'authentification.

```
aws iam resync-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 987654
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Utilisation de l'authentification multifactorielle (MFA) dans AWS dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ResyncMfaDeviceà la section Référence des AWS CLI commandes.

`set-default-policy-version`

L'exemple de code suivant montre comment utiliser`set-default-policy-version`.

AWS CLI

Pour définir la version spécifiée de la politique spécifiée comme la version par défaut de la politique.

Cet exemple définit la version `v2` de la politique dont l'ARN est `arn:aws:iam::123456789012:policy/MyPolicy` en tant que version active par défaut.

```
aws iam set-default-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Pour de plus amples informations, veuillez consulter Policies and permissions in IAM (Stratégies et autorisations dans IAM) dans le AWS IAM Guide de l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous SetDefaultPolicyVersionà la section Référence des AWS CLI commandes.

`set-security-token-service-preferences`

L'exemple de code suivant montre comment utiliser`set-security-token-service-preferences`.

AWS CLI

Pour définir la version du jeton de point de terminaison global

L'`set-security-token-service-preferences`exemple suivant configure Amazon STS pour utiliser les jetons de version 2 lorsque vous vous authentifiez auprès du point de terminaison global.

```
aws iam set-security-token-service-preferences \
    --global-endpoint-token-version v2Token
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion du AWS STS dans une AWS région dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous SetSecurityTokenServicePreferencesà la section Référence des AWS CLI commandes.

`simulate-custom-policy`

L'exemple de code suivant montre comment utiliser`simulate-custom-policy`.

AWS CLI

Exemple 1 : pour simuler les effets de toutes les politiques IAM associées à un utilisateur ou à un rôle IAM

Ce qui suit `simulate-custom-policy` montre comment fournir à la fois la politique, définir les valeurs des variables et simuler un appel d'API pour voir s'il est autorisé ou refusé. L'exemple suivant montre une politique qui permet l'accès à la base de données uniquement après une date et une heure spécifiées. La simulation réussit car les actions simulées et la `aws:CurrentTime` variable spécifiée répondent toutes aux exigences de la politique.

```
aws iam simulate-custom-policy \
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}}' \
    --action-names dynamodb:CreateBackup \
    --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2019-04-25T11:00:00Z',ContextKey
```

Sortie :

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "allowed",
            "MatchedStatements": [
                {
                    "SourcePolicyId": "PolicyInputList.1",
                    "StartPosition": {
                        "Line": 1,
                        "Column": 38
                    },
```

```
                    "EndPosition": {
                        "Line": 1,
                        "Column": 167
                    }
                }
            ],
            "MissingContextValues": []
        }
    ]
}
```

Exemple 2 : pour simuler une commande interdite par la politique

L'`simulate-custom-policy`exemple suivant montre les résultats de la simulation d'une commande interdite par la politique. Dans cet exemple, la date fournie est antérieure à celle requise par la condition de la police.

```
aws iam simulate-custom-policy \
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}}' \
    --action-names dynamodb:CreateBackup \
    --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2014-04-25T11:00:00Z',ContextKey
```

Sortie :

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "implicitDeny",
            "MatchedStatements": [],
            "MissingContextValues": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Tester les politiques IAM à l'aide du simulateur de politiques IAM dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous SimulateCustomPolicyà la section Référence des AWS CLI commandes.

`simulate-principal-policy`

L'exemple de code suivant montre comment utilisersimulate-principal-policy.

AWS CLI

Exemple 1 : pour simuler les effets d'une politique IAM arbitraire

Ce qui suit `simulate-principal-policy` montre comment simuler un utilisateur appelant une action d'API et déterminer si les politiques associées à cet utilisateur autorisent ou refusent l'action. Dans l'exemple suivant, l'utilisateur dispose d'une politique qui autorise uniquement l'`codecommit:ListRepositories`action.

```
aws iam simulate-principal-policy \
    --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
    --action-names codecommit:ListRepositories
```

Sortie :

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "codecommit:ListRepositories",
            "EvalResourceName": "*",
            "EvalDecision": "allowed",
            "MatchedStatements": [
                {
                    "SourcePolicyId": "Grant-Access-To-CodeCommit-ListRepo",
                    "StartPosition": {
                        "Line": 3,
                        "Column": 19
                    },
                    "EndPosition": {
                        "Line": 9,
                        "Column": 10
                    }
                }
            ],
```

```
            "MissingContextValues": []
        }
    ]
}
```

Exemple 2 : Pour simuler les effets d'une commande interdite

L'`simulate-custom-policy`exemple suivant montre les résultats de la simulation d'une commande interdite par l'une des politiques de l'utilisateur. Dans l'exemple suivant, l'utilisateur dispose d'une politique qui autorise l'accès à une base de données DynamoDB uniquement après une certaine date et heure. Dans le cadre de la simulation, l'utilisateur tente d'accéder à la base de données avec une `aws:CurrentTime` valeur antérieure à celle autorisée par les conditions de la politique.

```
aws iam simulate-principal-policy \
    --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
    --action-names dynamodb:CreateBackup \
    --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2018-04-25T11:00:00Z',ContextKey
```

Sortie :

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "implicitDeny",
            "MatchedStatements": [],
            "MissingContextValues": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Tester les politiques IAM à l'aide du simulateur de politiques IAM dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous SimulatePrincipalPolicyà la section Référence des AWS CLI commandes.

`tag-instance-profile`

L'exemple de code suivant montre comment utiliser`tag-instance-profile`.

AWS CLI

Pour ajouter une balise à un profil d'instance

La `tag-instance-profile` commande suivante ajoute une balise avec un nom de département au profil d'instance spécifié.

```
aws iam tag-instance-profile \
    --instance-profile-name deployment-role \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagInstanceProfileà la section Référence des AWS CLI commandes.

`tag-mfa-device`

L'exemple de code suivant montre comment utiliser`tag-mfa-device`.

AWS CLI

Pour ajouter une étiquette à un appareil MFA

La `tag-mfa-device` commande suivante ajoute une balise portant le nom du département au périphérique MFA spécifié.

```
aws iam tag-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/alice \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagMfaDeviceà la section Référence des AWS CLI commandes.

`tag-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`tag-open-id-connect-provider`.

AWS CLI

Pour ajouter un tag à un fournisseur d'identité compatible avec OpenID Connect (OIDC)

La `tag-open-id-connect-provider` commande suivante ajoute une balise avec un nom de département au fournisseur d'identité OIDC spécifié.

```
aws iam tag-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`tag-policy`

L'exemple de code suivant montre comment utiliser`tag-policy`.

AWS CLI

Pour ajouter un tag à une politique gérée par le client

La `tag-policy` commande suivante ajoute une balise avec un nom de département à la politique gérée par le client spécifiée.

```
aws iam tag-policy \
    --policy-arn arn:aws:iam::123456789012:policy/billing-access \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagPolicyà la section Référence des AWS CLI commandes.

`tag-role`

L'exemple de code suivant montre comment utiliser`tag-role`.

AWS CLI

Pour ajouter une balise à un rôle

La commande `tag-role` suivante ajoute une balise avec un nom de département au rôle spécifié.

```
aws iam tag-role --role-name my-role \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagRoleà la section Référence des AWS CLI commandes.

`tag-saml-provider`

L'exemple de code suivant montre comment utiliser`tag-saml-provider`.

AWS CLI

Pour ajouter une balise à un fournisseur SAML

La `tag-saml-provider` commande suivante ajoute une balise avec un nom de département au fournisseur SAML spécifié.

```
aws iam tag-saml-provider \
```

```
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagSamlProviderà la section Référence des AWS CLI commandes.

`tag-server-certificate`

L'exemple de code suivant montre comment utilisertag-server-certificate.

AWS CLI

Pour ajouter une balise à un certificat de serveur

La `tag-saml-provider` commande suivante ajoute une balise avec un nom de département au certificat de serveur spécifié.

```
aws iam tag-server-certificate \
    --server-certificate-name ExampleCertificate \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagServerCertificateà la section Référence des AWS CLI commandes.

`tag-user`

L'exemple de code suivant montre comment utilisertag-user.

AWS CLI

Pour ajouter une balise à un utilisateur

La commande `tag-user` suivante ajoute une balise avec le département associé à l'utilisateur spécifié.

```
aws iam tag-user \
    --user-name alice \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagUserà la section Référence des AWS CLI commandes.

`untag-instance-profile`

L'exemple de code suivant montre comment utiliser`untag-instance-profile`.

AWS CLI

Pour supprimer une balise d'un profil d'instance

La `untag-instance-profile` commande suivante supprime toute balise portant le nom de clé « Department » du profil d'instance spécifié.

```
aws iam untag-instance-profile \
    --instance-profile-name deployment-role \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagInstanceProfileà la section Référence des AWS CLI commandes.

`untag-mfa-device`

L'exemple de code suivant montre comment utiliser`untag-mfa-device`.

AWS CLI

Pour supprimer une étiquette d'un appareil MFA

La `untag-mfa-device` commande suivante supprime toute balise portant le nom de clé « Department » du périphérique MFA spécifié.

```
aws iam untag-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/alice \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagMfaDeviceà la section Référence des AWS CLI commandes.

`untag-open-id-connect-provider`

L'exemple de code suivant montre comment utiliser`untag-open-id-connect-provider`.

AWS CLI

Pour supprimer une étiquette d'un fournisseur d'identité OIDC

La `untag-open-id-connect-provider` commande suivante supprime toute balise portant le nom de clé « Department » du fournisseur d'identité OIDC spécifié.

```
aws iam untag-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagOpenIdConnectProviderà la section Référence des AWS CLI commandes.

`untag-policy`

L'exemple de code suivant montre comment utiliser`untag-policy`.

AWS CLI

Pour supprimer un tag d'une politique gérée par le client

La `untag-policy` commande suivante supprime toute balise portant le nom de clé « Department » de la politique gérée par le client spécifiée.

```
aws iam untag-policy \
    --policy-arn arn:aws:iam::452925170507:policy/billing-access \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagPolicyà la section Référence des AWS CLI commandes.

`untag-role`

L'exemple de code suivant montre comment utiliser`untag-role`.

AWS CLI

Pour supprimer une balise d'un rôle

La commande `untag-role` suivante supprime toute balise portant le nom de clé « Département » du rôle spécifié.

```
aws iam untag-role \
    --role-name my-role \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagRoleà la section Référence des AWS CLI commandes.

`untag-saml-provider`

L'exemple de code suivant montre comment utiliser`untag-saml-provider`.

AWS CLI

Pour supprimer une balise d'un fournisseur SAML

La `untag-saml-provider` commande suivante supprime toute balise portant le nom de clé « Department » du profil d'instance spécifié.

```
aws iam untag-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagSamlProviderà la section Référence des AWS CLI commandes.

`untag-server-certificate`

L'exemple de code suivant montre comment utiliser`untag-server-certificate`.

AWS CLI

Pour supprimer une étiquette d'un certificat de serveur

La `untag-server-certificate` commande suivante supprime toute balise portant le nom de clé « Department » du certificat de serveur spécifié.

```
aws iam untag-server-certificate \
    --server-certificate-name ExampleCertificate \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagServerCertificateà la section Référence des AWS CLI commandes.

`untag-user`

L'exemple de code suivant montre comment utiliser`untag-user`.

AWS CLI

Pour supprimer une balise d'un utilisateur

La commande `untag-user` suivante supprime toute balise portant le nom de clé « Department » de l'utilisateur spécifié.

```
aws iam untag-user \
    --user-name alice \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, veuillez consulter Étiquetage de ressources IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagUserà la section Référence des AWS CLI commandes.

`update-access-key`

L'exemple de code suivant montre comment utiliser`update-access-key`.

AWS CLI

Pour activer ou désactiver une clé d'accès pour un utilisateur IAM

La commande `update-access-key` suivante désactive la clé d'accès spécifiée (un ID de clé d'accès rapide et une clé d'accès secrète) pour l'utilisateur IAM nommé Bob.

```
aws iam update-access-key \
    --access-key-id AKIAIOSFODNN7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

Cette commande ne produit aucun résultat.

La désactivation de la clé signifie qu'elle ne peut pas être utilisée pour un accès programmatique à. AWS Cependant, la clé est toujours disponible et peut être réactivée.

Pour de plus amples informations, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateAccessKeyà la section Référence des AWS CLI commandes.

`update-account-password-policy`

L'exemple de code suivant montre comment utiliser`update-account-password-policy`.

AWS CLI

Pour définir ou modifier la politique de mot de passe du compte actuel

La commande `update-account-password-policy` suivante définit la politique de mot de passe pour exiger une longueur minimale de huit caractères et un ou plusieurs chiffres dans le mot de passe.

```
aws iam update-account-password-policy \
    --minimum-password-length 8 \
    --require-numbers
```

Cette commande ne produit aucun résultat.

Les modifications apportées à la politique de mot de passe d'un compte affectent tous les nouveaux mots de passe créés pour les utilisateurs IAM du compte. Les modifications apportées à la politique des mots de passe n'affectent pas les mots de passe existants.

Pour plus d'informations, consultez la section Définition d'une politique de mot de passe du compte pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateAccountPasswordPolicyà la section Référence des AWS CLI commandes.

`update-assume-role-policy`

L'exemple de code suivant montre comment utiliser`update-assume-role-policy`.

AWS CLI

Pour mettre à jour la politique d'approbation d'un rôle IAM

La commande `update-assume-role-policy` suivante met à jour la politique d'approbation pour le rôle nommé `Test-Role`.

```
aws iam update-assume-role-policy \
    --role-name Test-Role \
    --policy-document file://Test-Role-Trust-Policy.json
```

Cette commande ne produit aucun résultat.

La politique d'approbation est définie sous la forme d'un document JSON dans le fichier Test-Role-Trust-Policy.json. (Le nom et l'extension du fichier n'ont aucune importance.) La politique d'approbation doit spécifier un principal.

Pour mettre à jour la politique des autorisations d'un rôle, utilisez la commande `put-role-policy`.

Pour plus d'informations, consultez Création de rôles IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateAssumeRolePolicyà la section Référence des AWS CLI commandes.

`update-group`

L'exemple de code suivant montre comment utiliser`update-group`.

AWS CLI

Pour renommer un groupe IAM

La commande `update-group` suivante remplace le nom du groupe IAM `Test` par `Test-1`.

```
aws iam update-group \
    --group-name Test \
    --new-group-name Test-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Affectation d'un nouveau nom à un groupe IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupà la section Référence des AWS CLI commandes.

`update-login-profile`

L'exemple de code suivant montre comment utiliser`update-login-profile`.

AWS CLI

Pour mettre à jour le mot de passe d'un utilisateur IAM

Par exemple, la commande suivante crée un utilisateur IAM nommé `update-login-profile` avec le mot de passe Bob.

```
aws iam update-login-profile \
    --user-name Bob \
    --password <password>
```

Cette commande ne produit aucun résultat.

Pour définir une politique de mot de passe pour le compte, utilisez la commande `update-account-password-policy`. Si le nouveau mot de passe enfreint la politique de mot de passe du compte, la commande renvoie une erreur `PasswordPolicyViolation`.

Si la politique de mot de passe du compte le permet, les utilisateurs IAM peuvent modifier leurs propres mots de passe à l'aide de la commande `change-password`.

Conservez le nouveau mot de passe en lieu sûr. Si le mot de passe est perdu, il ne peut pas être récupéré et vous devez recréer un à l'aide de la commande `create-login-profile`.

Pour de plus amples informations, consultez Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateLoginProfileà la section Référence des AWS CLI commandes.

`update-open-id-connect-provider-thumbprint`

L'exemple de code suivant montre comment utiliserupdate-open-id-connect-provider-thumbprint.

AWS CLI

Pour remplacer la liste existante des empreintes numériques des certificats de serveur par une nouvelle liste

Cet exemple met à jour la liste des empreintes numériques des certificats pour le fournisseur OIDC dont l'ARN est `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` afin d'utiliser une nouvelle empreinte.

```
aws iam update-open-id-connect-provider-thumbprint \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com \
    --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Création de fournisseurs d'identité OpenID Connect (OIDC) du Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateOpenIdConnectProviderThumbprintà la section Référence des AWS CLI commandes.

`update-role-description`

L'exemple de code suivant montre comment utiliserupdate-role-description.

AWS CLI

Pour modifier la description d'un rôle IAM

La commande `update-role` suivante remplace le nom du rôle IAM `production-role` par `Main production role`.

```
aws iam update-role-description \
    --role-name production-role \
    --description 'Main production role'
```

Sortie :

```
{
    "Role": {
        "Path": "/",
        "RoleName": "production-role",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/production-role",
        "CreateDate": "2017-12-06T17:16:37+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole",
                    "Condition": {}
                }
            ]
        },
        "Description": "Main production role"
    }
}
```

Pour plus d'informations, consultez Modification d'un rôle dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateRoleDescriptionà la section Référence des AWS CLI commandes.

`update-role`

L'exemple de code suivant montre comment utiliser`update-role`.

AWS CLI

Pour modifier la description ou la durée de session d'un rôle IAM

La commande `update-role` suivante remplace la description du rôle IAM `production-role` par `Main production role` et définit la durée maximale de session à 12 heures.

```
aws iam update-role \
    --role-name production-role \
    --description 'Main production role' \
    --max-session-duration 43200
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modification d'un rôle dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateRoleà la section Référence des AWS CLI commandes.

`update-saml-provider`

L'exemple de code suivant montre comment utiliser`update-saml-provider`.

AWS CLI

Pour mettre à jour le document de métadonnées d'un fournisseur SAML existant

Cet exemple met à jour le fournisseur SAML dans IAM, dont l'ARN est `arn:aws:iam::123456789012:saml-provider/SAMLADFS`, avec un nouveau document de métadonnées SAML issu du fichier `SAMLMetaData.xml`.

```
aws iam update-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Sortie :

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

Pour plus d'informations, consultez Création de fournisseurs d'identité SAML IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateSamlProviderà la section Référence des AWS CLI commandes.

`update-server-certificate`

L'exemple de code suivant montre comment utiliser`update-server-certificate`.

AWS CLI

Pour modifier le chemin ou le nom d'un certificat de serveur dans votre AWS compte

La commande `update-server-certificate` suivante fait passer le nom du certificat de `myServerCertificate` à `myUpdatedServerCertificate`. Il modifie également le chemin pour `/cloudfront/` qu'il soit accessible par le CloudFront service Amazon. Cette commande ne produit aucun résultat. Vous pouvez afficher les résultats de la mise à jour en exécutant la commande `list-server-certificates`.

```
aws-iam update-server-certificate \
    --server-certificate-name myServerCertificate \
    --new-server-certificate-name myUpdatedServerCertificate \
    --new-path /cloudfront/
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, veuillez consulter Gestion des certificats de serveur dans IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateServerCertificateà la section Référence des AWS CLI commandes.

`update-service-specific-credential`

L'exemple de code suivant montre comment utiliser`update-service-specific-credential`.

AWS CLI

Exemple 1 : pour mettre à jour le statut des informations d'identification spécifiques au service de l'utilisateur demandeur

L'`update-service-specific-credential`exemple suivant modifie le statut des informations d'identification spécifiées pour l'utilisateur à qui la demande est adresséeInactive.

```
aws iam update-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
    --status Inactive
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour mettre à jour le statut des informations d'identification spécifiques au service d'un utilisateur spécifié

L'`update-service-specific-credential`exemple suivant fait passer le statut des informations d'identification de l'utilisateur spécifié à Inactif.

```
aws iam update-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
    --status Inactive
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer des informations d'identification Git pour les connexions HTTPS CodeCommit dans le guide de AWS CodeCommit l'utilisateur

- Pour plus de détails sur l'API, reportez-vous UpdateServiceSpecificCredentialà la section Référence des AWS CLI commandes.

`update-signing-certificate`

L'exemple de code suivant montre comment utiliser`update-signing-certificate`.

AWS CLI

Pour activer ou désactiver un certificat de signature pour un utilisateur IAM

La commande `update-signing-certificate` suivante désactive le certificat de signature spécifié pour l'utilisateur IAM nommé Bob.

```
aws iam update-signing-certificate \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

Pour obtenir l'ID d'un certificat de signature, utilisez la commande `list-signing-certificates`.

Pour plus d'informations, consultez la section Gérer les certificats de signature dans le guide de EC2 l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateSigningCertificateà la section Référence des AWS CLI commandes.

`update-ssh-public-key`

L'exemple de code suivant montre comment utiliser`update-ssh-public-key`.

AWS CLI

Pour modifier le statut d'une clé publique SSH

La `update-ssh-public-key` commande suivante change le statut de la clé publique spécifiée en`Inactive`.

```
aws iam update-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA1234567890EXAMPLE \
    --status Inactive
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser les clés SSH et SSH avec CodeCommit dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateSshPublicKeyà la section Référence des AWS CLI commandes.

`update-user`

L'exemple de code suivant montre comment utiliser`update-user`.

AWS CLI

Pour modifier le nom d’un utilisateur IAM

La commande `update-user` suivante fait passer le nom de l’utilisateur IAM de `Bob` à `Robert`.

```
aws iam update-user \
    --user-name Bob \
    --new-user-name Robert
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Affectation d'un nouveau nom à un groupe IAM dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateUserà la section Référence des AWS CLI commandes.

`upload-server-certificate`

L'exemple de code suivant montre comment utiliser`upload-server-certificate`.

AWS CLI

Pour télécharger un certificat de serveur sur votre AWS compte

La upload-server-certificatecommande suivante télécharge un certificat de serveur sur votre AWS compte. Dans cet exemple, le certificat se trouve dans le fichier `public_key_cert_file.pem`, la clé privée associée se trouve dans le fichier `my_private_key.pem` et la chaîne de certificats fournie par l'autorité de certification (CA) se trouve dans le fichier `my_certificate_chain_file.pem`. Lorsque le téléchargement du fichier est terminé, il est disponible sous le nom myServerCertificate. Les paramètres commençant par `file://` indiquent à la commande de lire le contenu du fichier et de l'utiliser comme valeur de paramètre au lieu du nom du fichier lui-même.

```
aws iam upload-server-certificate \
    --server-certificate-name myServerCertificate \
    --certificate-body file://public_key_cert_file.pem \
    --private-key file://my_private_key.pem \
    --certificate-chain file://my_certificate_chain_file.pem
```

Sortie :

```
{
    "ServerCertificateMetadata": {
        "Path": "/",
        "ServerCertificateName": "myServerCertificate",
        "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
        "Arn": "arn:aws:iam::1234567989012:server-certificate/myServerCertificate",
        "UploadDate": "2019-04-22T21:13:44+00:00",
        "Expiration": "2019-10-15T22:23:16+00:00"
    }
}
```

Pour de plus amples informations, veuillez consulter Creating, Uploading, and Deleting Server Certificates dans le guide d'utilisation d'IAM.

- Pour plus de détails sur l'API, reportez-vous UploadServerCertificateà la section Référence des AWS CLI commandes.

`upload-signing-certificate`

L'exemple de code suivant montre comment utiliser`upload-signing-certificate`.

AWS CLI

Pour charger un certificat de signature pour un utilisateur IAM

La commande `upload-signing-certificate` suivante charge un certificat de signature pour l'utilisateur IAM nommé Bob.

```
aws iam upload-signing-certificate \
    --user-name Bob \
    --certificate-body file://certificate.pem
```

Sortie :

```
{
    "Certificate": {
        "UserName": "Bob",
        "Status": "Active",
        "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END
 CERTIFICATE-----",
        "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
        "UploadDate": "2013-06-06T21:40:08.121Z"
    }
}
```

Le certificat se trouve dans un fichier au format PEM nommé certificate.pem.

Pour de plus amples informations, consultez Création, chargement et suppression de certificats de serveur dans le guide Utilisation d'IAM.

- Pour plus de détails sur l'API, reportez-vous UploadSigningCertificateà la section Référence des AWS CLI commandes.

`upload-ssh-public-key`

L'exemple de code suivant montre comment utiliser`upload-ssh-public-key`.

AWS CLI

Pour télécharger une clé publique SSH et l'associer à un utilisateur

La `upload-ssh-public-key` commande suivante télécharge la clé publique trouvée dans le fichier `sshkey.pub` et l'attache à l'utilisateur`sofia`.

```
aws iam upload-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-body file://sshkey.pub
```

Sortie :

```
{
    "SSHPublicKey": {
        "UserName": "sofia",
        "SSHPublicKeyId": "APKA1234567890EXAMPLE",
        "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
        "SSHPublicKeyBody": "ssh-rsa <<long string generated by ssh-keygen
 command>>",
        "Status": "Active",
        "UploadDate": "2019-04-18T17:04:49+00:00"
    }
}
```

Pour plus d'informations sur la façon de générer des clés dans un format adapté à cette commande, voir SSH et Linux, macOS ou Unix : configurer les clés publiques et privées pour Git et/ou SSH et CodeCommit Windows : configurer les clés publiques et privées pour Git et CodeCommit dans le Guide de l'AWS CodeCommit utilisateur.

- Pour plus de détails sur l'API, reportez-vous UploadSshPublicKeyà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'IAM Access Analyzer AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l' AWS Command Line Interface analyseur d'accès IAM.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`apply-archive-rule`

L'exemple de code suivant montre comment utiliser`apply-archive-rule`.

AWS CLI

Pour appliquer une règle d'archivage aux résultats existants qui répondent aux critères de la règle d'archivage

L'`apply-archive-rule`exemple suivant applique une règle d'archivage aux résultats existants qui répondent aux critères de la règle d'archivage.

```
aws accessanalyzer apply-archive-rule \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Règles d'archivage dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ApplyArchiveRuleà la section Référence des AWS CLI commandes.

`cancel-policy-generation`

L'exemple de code suivant montre comment utiliser`cancel-policy-generation`.

AWS CLI

Pour annuler la génération de politiques demandée

L'`cancel-policy-generation`exemple suivant annule l'identifiant de tâche de génération de politiques demandé.

```
aws accessanalyzer cancel-policy-generation \
    --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Génération de politiques de l'analyseur d'accès IAM dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous CancelPolicyGenerationà la section Référence des AWS CLI commandes.

`check-access-not-granted`

L'exemple de code suivant montre comment utiliser`check-access-not-granted`.

AWS CLI

Pour vérifier si l'accès spécifié n'est pas autorisé par une politique

L'`check-access-not-granted`exemple suivant vérifie si l'accès spécifié n'est pas autorisé par une politique.

```
aws accessanalyzer check-access-not-granted \
    --policy-document file://myfile.json \
    --access actions="s3:DeleteBucket","s3:GetBucketLocation" \
    --policy-type IDENTITY_POLICY
```

Contenu de `myfile.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Sortie :

```
{
    "result": "PASS",
    "message": "The policy document does not grant access to perform one or more of
 the listed actions."
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CheckAccessNotGrantedà la section Référence des AWS CLI commandes.

`check-no-new-access`

L'exemple de code suivant montre comment utiliser`check-no-new-access`.

AWS CLI

Pour vérifier si un nouvel accès est autorisé pour une politique mise à jour par rapport à la politique existante

L'`check-no-new-access`exemple suivant vérifie si un nouvel accès est autorisé pour une politique mise à jour par rapport à la politique existante.

```
aws accessanalyzer check-no-new-access \
    --existing-policy-document file://existing-policy.json \
    --new-policy-document file://new-policy.json \
    --policy-type IDENTITY_POLICY
```

Contenu de `existing-policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Contenu de `new-policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Sortie :

```
{
    "result": "FAIL",
```

```
    "message": "The modified permissions grant new access compared to your existing
 policy.",
    "reasons": [
        {
            "description": "New access in the statement with index: 0.",
            "statementIndex": 0
        }
    ]
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CheckNoNewAccessà la section Référence des AWS CLI commandes.

`check-no-public-access`

L'exemple de code suivant montre comment utiliser`check-no-public-access`.

AWS CLI

Pour vérifier si une politique de ressources peut accorder un accès public au type de ressource spécifié

L'`check-no-public-access`exemple suivant vérifie si une politique de ressources peut accorder un accès public au type de ressource spécifié.

```
aws accessanalyzer check-no-public-access \
    --policy-document file://check-no-public-access-myfile.json \
    --resource-type AWS::S3::Bucket
```

Contenu de `myfile.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CheckNoPublicAccess",
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam::111122223333:user/JohnDoe" },
            "Action": [
```

```
                "s3:GetObject"
            ]
        }
    ]
}
```

Sortie :

```
{
    "result": "PASS",
    "message": "The resource policy does not grant public access for the given
 resource type."
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CheckNoPublicAccessà la section Référence des AWS CLI commandes.

`create-access-preview`

L'exemple de code suivant montre comment utiliser`create-access-preview`.

AWS CLI

Pour créer un aperçu des accès qui vous permet de prévisualiser les résultats d'IAM Access Analyzer pour votre ressource avant de déployer les autorisations relatives aux ressources

L'`create-access-preview`exemple suivant crée un aperçu des accès qui vous permet de prévisualiser les résultats d'IAM Access Analyzer pour votre ressource avant de déployer les autorisations de ressources dans votre AWS compte.

```
aws accessanalyzer create-access-preview \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --configurations file://myfile.json
```

Contenu de `myfile.json` :

```
{
```

```
    "arn:aws:s3:::amzn-s3-demo-bucket": {
        "s3Bucket": {
            "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},\"Action
\":[\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":\"arn:aws:s3:::amzn-s3-demo-
bucket/*\"}]}",
            "bucketPublicAccessBlock": {
                "ignorePublicAcls": true,
                "restrictPublicBuckets": true
            },
            "bucketAclGrants": [
                {
                    "grantee": {
                        "id":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                    },
                    "permission": "READ"
                }
            ]
        }
    }
}
```

Sortie :

```
{
    "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateAccessPreviewà la section Référence des AWS CLI commandes.

`create-analyzer`

L'exemple de code suivant montre comment utiliser`create-analyzer`.

AWS CLI

Pour créer un analyseur

L'`create-analyzer`exemple suivant crée un analyseur dans votre AWS compte.

```
aws accessanalyzer create-analyzer \
    --analyzer-name example \
    --type ACCOUNT
```

Sortie :

```
{
    "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"
}
```

Pour plus d'informations, consultez les conclusions de Getting started with AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateAnalyzerà la section Référence des AWS CLI commandes.

`create-archive-rule`

L'exemple de code suivant montre comment utiliser`create-archive-rule`.

AWS CLI

Pour créer une règle d'archivage pour l'analyseur spécifié

L'`create-archive-rule`exemple suivant crée une règle d'archivage pour l'analyseur spécifié dans votre AWS compte.

```
aws accessanalyzer create-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyRule \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Règles d'archivage dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous CreateArchiveRuleà la section Référence des AWS CLI commandes.

`delete-analyzer`

L'exemple de code suivant montre comment utiliser`delete-analyzer`.

AWS CLI

Pour supprimer l'analyseur spécifié

L'`delete-analyzer`exemple suivant supprime l'analyseur spécifié dans votre AWS compte.

```
aws accessanalyzer delete-analyzer \
    --analyzer-name example
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Règles d'archivage dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteAnalyzerà la section Référence des AWS CLI commandes.

`delete-archive-rule`

L'exemple de code suivant montre comment utiliser`delete-archive-rule`.

AWS CLI

Pour supprimer la règle d'archivage spécifiée

L'`delete-archive-rule`exemple suivant supprime la règle d'archivage spécifiée dans votre AWS compte.

```
aws accessanalyzer delete-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyRule
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Règles d'archivage dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous DeleteArchiveRuleà la section Référence des AWS CLI commandes.

`get-access-preview`

L'exemple de code suivant montre comment utiliser`get-access-preview`.

AWS CLI

Pour récupérer des informations sur un aperçu des accès pour l'analyseur spécifié

L'`get-access-preview`exemple suivant permet de récupérer des informations sur un aperçu des accès pour l'analyseur spécifié dans votre AWS compte.

```
aws accessanalyzer get-access-preview \
    --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

Sortie :

```
{
    "accessPreview": {
        "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",
        "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
        "configurations": {
            "arn:aws:s3:::amzn-s3-demo-bucket": {
                "s3Bucket": {
                    "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},
\"Action\":[\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":\"arn:aws:s3:::amzn-
s3-demo-bucket/*\"}]}",
                    "bucketAclGrants": [
                        {
                            "permission": "READ",
                            "grantee": {
                                "id":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                            }
                        },
```

```
                    "bucketPublicAccessBlock": {
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    }
                }
            }
        },
        "createdAt": "2024-02-17T00:18:44+00:00",
        "status": "COMPLETED"
    }
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAccessPreviewà la section Référence des AWS CLI commandes.

`get-analyzed-resource`

L'exemple de code suivant montre comment utiliser`get-analyzed-resource`.

AWS CLI

Pour récupérer des informations sur une ressource analysée

L'`get-analyzed-resource`exemple suivant permet de récupérer des informations sur une ressource analysée dans votre AWS compte.

```
aws accessanalyzer get-analyzed-resource \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-arn arn:aws:s3:::amzn-s3-demo-bucket
```

Sortie :

```
{
    "resource": {
        "analyzedAt": "2024-02-15T18:01:53.002000+00:00",
        "isPublic": false,
        "resourceArn": "arn:aws:s3:::amzn-s3-demo-bucket",
```

```
        "resourceOwnerAccount": "111122223333",
        "resourceType": "AWS::S3::Bucket"
    }
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAnalyzedResourceà la section Référence des AWS CLI commandes.

`get-analyzer`

L'exemple de code suivant montre comment utiliser`get-analyzer`.

AWS CLI

Pour récupérer des informations sur l'analyseur spécifié

L'`get-analyzer`exemple suivant permet de récupérer des informations sur l'analyseur spécifié dans votre AWS compte.

```
aws accessanalyzer get-analyzer \
    --analyzer-name ConsoleAnalyzer-account
```

Sortie :

```
{
    "analyzer": {
        "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
        "createdAt": "2019-12-03T07:28:17+00:00",
        "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-topic",
        "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
        "name": "ConsoleAnalyzer-account",
        "status": "ACTIVE",
        "tags": {
            "auto-delete": "no"
        },
        "type": "ACCOUNT"
    }
```

```
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetAnalyzerà la section Référence des AWS CLI commandes.

`get-archive-rule`

L'exemple de code suivant montre comment utiliser`get-archive-rule`.

AWS CLI

Pour récupérer des informations relatives à une règle d'archivage

L'`get-archive-rule`exemple suivant permet de récupérer des informations sur une règle d'archivage de votre AWS compte.

```
aws accessanalyzer get-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule
```

Sortie :

```
{
    "archiveRule": {
        "createdAt": "2024-02-15T00:49:27+00:00",
        "filter": {
            "resource": {
                "contains": [
                    "Cognito"
                ]
            },
            "resourceType": {
                "eq": [
                    "AWS::IAM::Role"
                ]
            }
        },
        "ruleName": "MyArchiveRule",
```

```
        "updatedAt": "2024-02-15T00:49:27+00:00"
    }
}
```

Pour plus d'informations, consultez la section Règles d'archivage dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetArchiveRuleà la section Référence des AWS CLI commandes.

`get-finding-v2`

L'exemple de code suivant montre comment utiliser`get-finding-v2`.

AWS CLI

Pour récupérer des informations sur le résultat spécifié

L'`get-finding-v2`exemple suivant permet de récupérer des informations sur le résultat spécifié dans votre AWS compte.

```
aws accessanalyzer get-finding-v2 \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
    --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

Sortie :

```
{
    "findingDetails": [
        {
            "externalAccessDetails": {
                "action": [
                    "sts:AssumeRoleWithWebIdentity"
                ],
                "condition": {
                    "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
                },
                "isPublic": false,
                "principal": {
```

```
                    "Federated": "cognito-identity.amazonaws.com"
                }
            }
        }
    ],
    "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
    "status": "ACTIVE",
    "error": null,
    "createdAt": "2021-02-26T21:17:50.905000+00:00",
    "resourceType": "AWS::IAM::Role",
    "findingType": "ExternalAccess",
    "resourceOwnerAccount": "111122223333",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
    "updatedAt": "2021-02-26T21:17:50.905000+00:00"
}
```

Pour plus d'informations, consultez la section Révision des résultats dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, voir GetFindingV2 dans AWS CLI Command Reference.

`get-finding`

L'exemple de code suivant montre comment utiliser`get-finding`.

AWS CLI

Pour récupérer des informations sur le résultat spécifié

L'`get-finding`exemple suivant permet de récupérer des informations sur le résultat spécifié dans votre AWS compte.

```
aws accessanalyzer get-finding \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
    --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

Sortie :

```
{
```

```
    "finding": {
        "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
        "principal": {
            "Federated": "cognito-identity.amazonaws.com"
        },
        "action": [
            "sts:AssumeRoleWithWebIdentity"
        ],
        "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
        "isPublic": false,
        "resourceType": "AWS::IAM::Role",
        "condition": {
            "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
        },
        "createdAt": "2021-02-26T21:17:50.905000+00:00",
        "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
        "updatedAt": "2021-02-26T21:17:50.905000+00:00",
        "status": "ACTIVE",
        "resourceOwnerAccount": "111122223333"
    }
}
```

Pour plus d'informations, consultez la section Révision des résultats dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetFindingà la section Référence des AWS CLI commandes.

`get-generated-policy`

L'exemple de code suivant montre comment utiliser`get-generated-policy`.

AWS CLI

Pour récupérer la politique générée à l'aide de l'API StartPolicyGeneration ``

L'`get-generated-policy`exemple suivant récupère la politique qui a été générée à l'aide de l' StartPolicyGeneration API de votre AWS compte.

```
aws accessanalyzer get-generated-policy \
    --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

Sortie :


```
{
    "generatedPolicyResult": {
        "generatedPolicies": [
            {
                "policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Sid\":\"SupportedServiceSid0\",\"Effect\":\"Allow\",\"Action\":
[\"access-analyzer:GetAnalyzer\",\"access-analyzer:ListAnalyzers\",
\"access-analyzer:ListArchiveRules\",\"access-analyzer:ListFindings
\",\"cloudtrail:DescribeTrails\",\"cloudtrail:GetEventDataStore\",
\"cloudtrail:GetEventSelectors\",\"cloudtrail:GetInsightSelectors
\",\"cloudtrail:GetTrailStatus\",\"cloudtrail:ListChannels\",
\"cloudtrail:ListEventDataStores\",\"cloudtrail:ListQueries\",\"cloudtrail:ListTags
\",\"cloudtrail:LookupEvents\",\"ec2:DescribeRegions\",\"iam:GetAccountSummary
\",\"iam:GetOpenIDConnectProvider\",\"iam:GetRole\",\"iam:ListAccessKeys\",
\"iam:ListAccountAliases\",\"iam:ListOpenIDConnectProviders\",\"iam:ListRoles
\",\"iam:ListSAMLProviders\",\"kms:ListAliases\",\"s3:GetBucketLocation\",
\"s3:ListAllMyBuckets\"],\"Resource\":\"*\"}]}"
            }
        ],
        "properties": {
            "cloudTrailProperties": {
                "endTime": "2024-02-14T22:44:40+00:00",
                "startTime": "2024-02-13T00:30:00+00:00",
                "trailProperties": [
                    {
                        "allRegions": true,
                        "cloudTrailArn": "arn:aws:cloudtrail:us-
west-2:111122223333:trail/my-trail",
                        "regions": []
                    }
                ]
            },
            "isComplete": false,
            "principalArn": "arn:aws:iam::111122223333:role/Admin"
        }
    },
    "jobDetails": {
        "completedOn": "2024-02-14T22:47:01+00:00",
        "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
        "startedOn": "2024-02-14T22:44:41+00:00",
        "status": "SUCCEEDED"
    }
```

```
}
```

Pour plus d'informations, consultez la section Génération de politiques de l'analyseur d'accès IAM dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous GetGeneratedPolicyà la section Référence des AWS CLI commandes.

`list-access-preview-findings`

L'exemple de code suivant montre comment utiliser`list-access-preview-findings`.

AWS CLI

Pour récupérer une liste des résultats d'aperçu d'accès générés par l'aperçu d'accès spécifié

L'`list-access-preview-findings`exemple suivant extrait une liste des résultats d'aperçu d'accès générés par l'aperçu d'accès spécifié dans votre AWS compte.

```
aws accessanalyzer list-access-preview-findings \
    --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

Sortie :

```
{
    "findings": [
        {
            "id": "e22fc158-1c87-4c32-9464-e7f405ce8d74",
            "principal": {
                "AWS": "111122223333"
            },
            "action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "condition": {},
            "resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "isPublic": false,
            "resourceType": "AWS::S3::Bucket",
            "createdAt": "2024-02-17T00:18:46+00:00",
```

```
            "changeType": "NEW",
            "status": "ACTIVE",
            "resourceOwnerAccount": "111122223333",
            "sources": [
                {
                    "type": "POLICY"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAccessPreviewFindingsà la section Référence des AWS CLI commandes.

`list-access-previews`

L'exemple de code suivant montre comment utiliser`list-access-previews`.

AWS CLI

Pour récupérer la liste des aperçus d'accès pour l'analyseur spécifié

L'`list-access-previews`exemple suivant extrait une liste d'aperçus d'accès pour l'analyseur spécifié dans votre compte. AWS

```
aws accessanalyzer list-access-previews \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

Sortie :

```
{
    "accessPreviews": [
        {
            "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",
            "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
            "createdAt": "2024-02-17T00:18:44+00:00",
```

```
            "status": "COMPLETED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Prévisualisation de l'accès avec IAM Access Analyzer APIs dans le guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAccessPreviewsà la section Référence des AWS CLI commandes.

`list-analyzed-resources`

L'exemple de code suivant montre comment utiliser`list-analyzed-resources`.

AWS CLI

Pour répertorier les widgets disponibles

L'`list-analyzed-resources`exemple suivant répertorie les widgets disponibles dans votre AWS compte.

```
aws accessanalyzer list-analyzed-resources \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-type AWS::IAM::Role
```

Sortie :

```
{
    "analyzedResources": [
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:Validation-Email",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        },
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:admin-alerts",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        },
        {
```

```
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:config-topic",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        },
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:inspector-topic",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        }
    ]
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAnalyzedResourcesà la section Référence des AWS CLI commandes.

`list-analyzers`

L'exemple de code suivant montre comment utiliser`list-analyzers`.

AWS CLI

Pour récupérer une liste d'analyseurs

L'`list-analyzers`exemple suivant permet de récupérer la liste des analyseurs de votre AWS compte.

```
aws accessanalyzer list-analyzers
```

Sortie :

```
{
    "analyzers": [
        {
            "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization",
            "createdAt": "2024-02-15T00:46:40+00:00",
            "name": "UnusedAccess-ConsoleAnalyzer-organization",
            "status": "ACTIVE",
            "tags": {
```

```
                "auto-delete": "no"
            },
            "type": "ORGANIZATION_UNUSED_ACCESS"
        },
        {
            "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization",
            "createdAt": "2020-04-25T07:43:28+00:00",
            "lastResourceAnalyzed": "arn:aws:s3:::amzn-s3-demo-bucket",
            "lastResourceAnalyzedAt": "2024-02-15T21:51:56.517000+00:00",
            "name": "ConsoleAnalyzer-organization",
            "status": "ACTIVE",
            "tags": {
                "auto-delete": "no"
            },
            "type": "ORGANIZATION"
        },
        {
            "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
            "createdAt": "2019-12-03T07:28:17+00:00",
            "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-
topic",
            "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
            "name": "ConsoleAnalyzer-account",
            "status": "ACTIVE",
            "tags": {
                "auto-delete": "no"
            },
            "type": "ACCOUNT"
        }
    ]
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListAnalyzersà la section Référence des AWS CLI commandes.

`list-archive-rules`

L'exemple de code suivant montre comment utiliser`list-archive-rules`.

AWS CLI

Pour récupérer la liste des règles d'archivage créées pour l'analyseur spécifié

L'`list-archive-rules`exemple suivant extrait une liste de règles d'archivage créées pour l'analyseur spécifié dans votre AWS compte.

```
aws accessanalyzer list-archive-rules \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization
```

Sortie :

```
{
    "archiveRules": [
        {
            "createdAt": "2024-02-15T00:49:27+00:00",
            "filter": {
                "resource": {
                    "contains": [
                        "Cognito"
                    ]
                },
                "resourceType": {
                    "eq": [
                        "AWS::IAM::Role"
                    ]
                }
            },
            "ruleName": "MyArchiveRule",
            "updatedAt": "2024-02-15T00:49:27+00:00"
        },
        {
            "createdAt": "2024-02-15T23:27:45+00:00",
            "filter": {
                "findingType": {
                    "eq": [
                        "UnusedIAMUserAccessKey"
                    ]
                }
            },
            "ruleName": "ArchiveRule-56125a39-e517-4ff8-afb1-ef06f58db612",
            "updatedAt": "2024-02-15T23:27:45+00:00"
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListArchiveRulesà la section Référence des AWS CLI commandes.

`list-findings-v2`

L'exemple de code suivant montre comment utiliser`list-findings-v2`.

AWS CLI

Pour récupérer une liste des résultats générés par l'analyseur spécifié

L'`list-findings-v2`exemple suivant extrait une liste des résultats générés par l'analyseur spécifié dans votre AWS compte. Cet exemple filtre les résultats pour inclure uniquement les rôles IAM dont le nom contient `Cognito`

```
aws accessanalyzer list-findings-v2 \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

Sortie :

```
{
    "findings": [
        {
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "createdAt": "2021-02-26T21:17:24.710000+00:00",
            "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
            "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
            "resourceType": "AWS::IAM::Role",
            "resourceOwnerAccount": "111122223333",
            "status": "ACTIVE",
            "updatedAt": "2021-02-26T21:17:24.710000+00:00",
            "findingType": "ExternalAccess"
```

```
        },
        {
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "createdAt": "2021-02-26T21:17:50.905000+00:00",
            "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
            "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
            "resourceType": "AWS::IAM::Role",
            "resourceOwnerAccount": "111122223333",
            "status": "ACTIVE",
            "updatedAt": "2021-02-26T21:17:50.905000+00:00",
            "findingType": "ExternalAccess"
        }
    ]
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, voir ListFindingsV2 dans AWS CLI Command Reference.

`list-findings`

L'exemple de code suivant montre comment utiliser`list-findings`.

AWS CLI

Pour récupérer une liste des résultats générés par l'analyseur spécifié

L'`list-findings`exemple suivant extrait une liste des résultats générés par l'analyseur spécifié dans votre AWS compte. Cet exemple filtre les résultats pour inclure uniquement les rôles IAM dont le nom contient `Cognito`

```
aws accessanalyzer list-findings \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

Sortie :

```
{
    "findings": [
        {
```

```
            "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
            "principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "action": [
                "sts:AssumeRoleWithWebIdentity"
            ],
            "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
            "isPublic": false,
            "resourceType": "AWS::IAM::Role",
            "condition": {
                "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
            },
            "createdAt": "2021-02-26T21:17:24.710000+00:00",
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "updatedAt": "2021-02-26T21:17:24.710000+00:00",
            "status": "ACTIVE",
            "resourceOwnerAccount": "111122223333"
        },
        {
            "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
            "principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "action": [
                "sts:AssumeRoleWithWebIdentity"
            ],
            "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
            "isPublic": false,
            "resourceType": "AWS::IAM::Role",
            "condition": {
                "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
            },
            "createdAt": "2021-02-26T21:17:50.905000+00:00",
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "updatedAt": "2021-02-26T21:17:50.905000+00:00",
            "status": "ACTIVE",
            "resourceOwnerAccount": "111122223333"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ListFindingsà la section Référence des AWS CLI commandes.

`list-policy-generations`

L'exemple de code suivant montre comment utiliser`list-policy-generations`.

AWS CLI

Pour répertorier toutes les générations de politiques demandées au cours des sept derniers jours

L'`list-policy-generations`exemple suivant répertorie toutes les générations de politiques demandées au cours des sept derniers jours sur votre AWS compte.

```
aws accessanalyzer list-policy-generations
```

Sortie :

```
{
    "policyGenerations": [
        {
            "completedOn": "2024-02-14T23:43:38+00:00",
            "jobId": "923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2",
            "principalArn": "arn:aws:iam::111122223333:role/Admin",
            "startedOn": "2024-02-14T23:43:02+00:00",
            "status": "CANCELED"
        },
        {
            "completedOn": "2024-02-14T22:47:01+00:00",
            "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
            "principalArn": "arn:aws:iam::111122223333:role/Admin",
            "startedOn": "2024-02-14T22:44:41+00:00",
            "status": "SUCCEEDED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Génération de politiques de l'analyseur d'accès IAM dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous ListPolicyGenerationsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour récupérer une liste de balises appliquées à la ressource spécifiée

L'`list-tags-for-resource`exemple suivant extrait une liste de balises appliquées à la ressource spécifiée dans votre AWS compte.

```
aws accessanalyzer list-tags-for-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

Sortie :

```
{
    "tags": {
        "Zone-of-trust": "Account",
        "Name": "ConsoleAnalyzer"
    }
}
```

Pour plus d'informations, consultez la section Génération de politiques de l'analyseur d'accès IAM dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-policy-generation`

L'exemple de code suivant montre comment utiliser`start-policy-generation`.

AWS CLI

Pour démarrer une demande de génération de politiques

L'`start-policy-generation`exemple suivant lance une demande de génération de politiques dans votre AWS compte.

```
aws accessanalyzer start-policy-generation \
    --policy-generation-details '{"principalArn":"arn:aws:iam::111122223333:role/
Admin"}' \
    --cloud-trail-details file://myfile.json
```

Contenu de `myfile.json` :

```
{
    "accessRole": "arn:aws:iam::111122223333:role/service-role/
AccessAnalyzerMonitorServiceRole",
    "startTime": "2024-02-13T00:30:00Z",
    "trails": [
        {
            "allRegions": true,
            "cloudTrailArn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/my-
trail"
        }
    ]
}
```

Sortie :

```
{
    "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"
}
```

Pour plus d'informations, consultez la section Génération de politiques de l'analyseur d'accès IAM dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous StartPolicyGenerationà la section Référence des AWS CLI commandes.

`start-resource-scan`

L'exemple de code suivant montre comment utiliser`start-resource-scan`.

AWS CLI

Pour lancer immédiatement une analyse des politiques appliquées à la ressource spécifiée

L'`start-resource-scan`exemple suivant lance immédiatement une analyse des politiques appliquées à la ressource spécifiée dans votre AWS compte.

```
aws accessanalyzer start-resource-scan \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-arn arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Génération de politiques de l'analyseur d'accès IAM dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous StartResourceScanà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter un tag à la ressource spécifiée

L'`tag-resource`exemple suivant ajoute une balise à la ressource spécifiée dans votre AWS compte.

```
aws accessanalyzer tag-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --tags Environment=dev,Purpose=testing
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les balises des ressources spécifiées

L'`untag-resource`exemple suivant supprime les balises de la ressource spécifiée dans votre AWS compte.

```
aws accessanalyzer untag-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --tag-keys Environment Purpose
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-archive-rule`

L'exemple de code suivant montre comment utiliser`update-archive-rule`.

AWS CLI

Pour mettre à jour les critères et les valeurs de la règle d'archivage spécifiée

L'`update-archive-rule`exemple suivant met à jour les critères et les valeurs de la règle d'archivage spécifiée dans votre AWS compte.

```
aws accessanalyzer update-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Règles d'archivage dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateArchiveRuleà la section Référence des AWS CLI commandes.

`update-findings`

L'exemple de code suivant montre comment utiliser`update-findings`.

AWS CLI

Pour mettre à jour le statut des résultats spécifiés

L'`update-findings`exemple suivant met à jour le statut des résultats spécifiés dans votre AWS compte.

```
aws accessanalyzer update-findings \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization \
    --ids 4f319ac3-2e0c-4dc4-bf51-7013a086b6ae 780d586a-2cce-4f72-aff6-359d450e7500
 \
    --status ARCHIVED
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Using AWS Identity and Access Management Access Analyzer dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous UpdateFindingsà la section Référence des AWS CLI commandes.

`validate-policy`

L'exemple de code suivant montre comment utiliser`validate-policy`.

AWS CLI

Pour demander la validation d'une politique et renvoyer une liste de résultats

L'`validate-policy`exemple suivant demande la validation d'une politique et renvoie une liste de résultats. La politique présentée dans cet exemple est une politique de confiance pour un rôle

Amazon Cognito utilisé pour la fédération d'identité Web. Les résultats générés par la politique de confiance concernent une valeur d'`Sid`élément vide et un principe de politique non concordant en raison de l'utilisation d'une action d'acceptation de rôle incorrecte. `sts:AssumeRole` L'action d'hypothèse de rôle correcte à utiliser avec Cognito est. `sts:AssumeRoleWithWebIdentity`

```
aws accessanalyzer validate-policy \
    --policy-document file://myfile.json \
    --policy-type RESOURCE_POLICY
```

Contenu de `myfile.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ],
            "Condition": {
                "StringEquals": {
                    "cognito-identity.amazonaws.com:aud": "us-west-2_EXAMPLE"
                }
            }
        }
    ]
}
```

Sortie :

```
{
    "findings": [
        {
            "findingDetails": "Add a value to the empty string in the Sid element.",
            "findingType": "SUGGESTION",
            "issueCode": "EMPTY_SID_VALUE",
```

```
            "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-suggestion-empty-sid-value",
            "locations": [
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Sid"
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 21,
                            "line": 5,
                            "offset": 81
                        },
                        "start": {
                            "column": 19,
                            "line": 5,
                            "offset": 79
                        }
                    }
                }
            ]
        },
        {
            "findingDetails": "The sts:AssumeRole action is invalid with the
 following principal(s): cognito-identity.amazonaws.com. Use a SAML provider
 principal with the sts:AssumeRoleWithSAML action or use an OIDC provider principal
 with the sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if
 you use either of the two options.",
            "findingType": "ERROR",
            "issueCode": "MISMATCHED_ACTION_FOR_PRINCIPAL",
            "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-mismatched-action-for-principal",
            "locations": [
                {
```

```
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Action"
                        },
                        {
                            "index": 0
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 32,
                            "line": 11,
                            "offset": 274
                        },
                        "start": {
                            "column": 16,
                            "line": 11,
                            "offset": 258
                        }
                    }
                },
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Principal"
                        },
                        {
                            "value": "Federated"
                        }
                    ],
                    "span": {
                        "end": {
```

```
                            "column": 61,
                            "line": 8,
                            "offset": 202
                        },
                        "start": {
                            "column": 29,
                            "line": 8,
                            "offset": 170
                        }
                    }
                }
            ]
        },
        {
            "findingDetails": "The following actions: sts:TagSession are not
 supported by the condition key cognito-identity.amazonaws.com:aud. The condition
 will not be evaluated for these actions. We recommend that you move these actions
 to a different statement without this condition key.",
            "findingType": "ERROR",
            "issueCode": "UNSUPPORTED_ACTION_FOR_CONDITION_KEY",
            "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-unsupported-action-for-condition-key",
            "locations": [
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Action"
                        },
                        {
                            "index": 1
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 32,
                            "line": 12,
                            "offset": 308
```

```
                        },
                        "start": {
                            "column": 16,
                            "line": 12,
                            "offset": 292
                        }
                    }
                },
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Condition"
                        },
                        {
                            "value": "StringEquals"
                        },
                        {
                            "value": "cognito-identity.amazonaws.com:aud"
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 79,
                            "line": 16,
                            "offset": 464
                        },
                        "start": {
                            "column": 58,
                            "line": 16,
                            "offset": 443
                        }
                    }
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Contrôles de validation des politiques dans le guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous ValidatePolicyà la section Référence des AWS CLI commandes.

Exemples d'Image Builder utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Image Builder.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-component`

L'exemple de code suivant montre comment utiliser`create-component`.

AWS CLI

Pour créer un composant

L'`create-component`exemple suivant crée un composant qui utilise un fichier de document JSON et fait référence à un document de composant au format YAML qui est chargé dans un compartiment Amazon S3.

```
aws imagebuilder create-component \
    --cli-input-json file://create-component.json
```

Contenu de `create-component.json` :

```
{
```

```
    "name": "MyExampleComponent",
    "semanticVersion": "2019.12.02",
    "description": "An example component that builds, validates and tests an image",
    "changeDescription": "Initial version.",
    "platform": "Windows",
    "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/examplecomponent/2019.12.02/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous CreateComponentà la section Référence des AWS CLI commandes.

`create-distribution-configuration`

L'exemple de code suivant montre comment utiliser`create-distribution-configuration`.

AWS CLI

Pour créer une configuration de distribution

L'`create-distribution-configuration`exemple suivant crée une configuration de distribution à l'aide d'un fichier JSON.

```
aws imagebuilder create-distribution-configuration \
    --cli-input-json file:/create-distribution-configuration.json
```

Contenu de `create-distribution-configuration.json` :

```
{
    "name": "MyExampleDistribution",
```

```
    "description": "Copies AMI to eu-west-1",
    "distributions": [
        {
            "region": "us-west-2",
            "amiDistributionConfiguration": {
                "name": "Name {{imagebuilder:buildDate}}",
                "description": "An example image name with parameter references",
                "amiTags": {
                    "KeyName": "{{ssm:parameter_name}}"
                },
                "launchPermission": {
                    "userIds": [
                        "123456789012"
                    ]
                }
            }
        },
        {
            "region": "eu-west-1",
            "amiDistributionConfiguration": {
                "name": "My {{imagebuilder:buildVersion}} image
 {{imagebuilder:buildDate}}",
                "amiTags": {
                    "KeyName": "Value"
                },
                "launchPermission": {
                    "userIds": [
                        "123456789012"
                    ]
                }
            }
        }
    ]
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous CreateDistributionConfigurationà la section Référence des AWS CLI commandes.

`create-image-pipeline`

L'exemple de code suivant montre comment utiliser`create-image-pipeline`.

AWS CLI

Pour créer un pipeline d'images

L'`create-image-pipeline`exemple suivant crée un pipeline d'images à l'aide d'un fichier JSON.

```
aws imagebuilder create-image-pipeline \
    --cli-input-json file://create-image-pipeline.json
```

Contenu de `create-image-pipeline.json` :

```
{
    "name": "MyWindows2016Pipeline",
    "description": "Builds Windows 2016 Images",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
    "imageTestsConfiguration": {
        "imageTestsEnabled": true,
        "timeoutMinutes": 60
    },
    "schedule": {
        "scheduleExpression": "cron(0 0 * * SUN)",
        "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "ENABLED"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous CreateImagePipelineà la section Référence des AWS CLI commandes.

`create-image-recipe`

L'exemple de code suivant montre comment utiliser`create-image-recipe`.

AWS CLI

Pour créer une recette

L'`create-image-recipe`exemple suivant crée une recette d'image à l'aide d'un fichier JSON. Les composants sont installés dans l'ordre dans lequel ils sont spécifiés.

```
aws imagebuilder create-image-recipe \
    --cli-input-json file://create-image-recipe.json
```

Contenu de `create-image-recipe.json` :

```
{
    "name": "MyBasicRecipe",
    "description": "This example image recipe creates a Windows 2016 image.",
    "semanticVersion": "2019.12.03",
    "components":
    [
        {
            "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1"
        },
```

```
        {
            "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myimportedcomponent/1.0.0/1"
        }
    ],
    "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-
english-full-base-x86/xxxx.x.x"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous CreateImageRecipeà la section Référence des AWS CLI commandes.

`create-image`

L'exemple de code suivant montre comment utiliser`create-image`.

AWS CLI

Pour créer une image

L'`create-image`exemple suivant crée une image.

```
aws imagebuilder create-image \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03 \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous CreateImageà la section Référence des AWS CLI commandes.

`create-infrastructure-configuration`

L'exemple de code suivant montre comment utiliser`create-infrastructure-configuration`.

AWS CLI

Pour créer une configuration d'infrastructure

L'`create-infrastructure-configuration`exemple suivant crée une configuration d'infrastructure à l'aide d'un fichier JSON.

```
aws imagebuilder create-infrastructure-configuration \
    --cli-input-json file://create-infrastructure-configuration.json
```

Contenu de `create-infrastructure-configuration.json` :

```
{
    "name": "MyExampleInfrastructure",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
        "m5.large", "m5.xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageBuilder",
    "securityGroupIds": [
        "sg-a1b2c3d4"
    ],
    "subnetId": "subnet-a1b2c3d4",
    "logging": {
```

```
        "s3Logs": {
            "s3BucketName": "bucket-name",
            "s3KeyPrefix": "bucket-path"
        }
    },
    "keyPair": "key-pair-name",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-topic-name"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous CreateInfrastructureConfigurationà la section Référence des AWS CLI commandes.

`delete-component`

L'exemple de code suivant montre comment utiliser`delete-component`.

AWS CLI

Pour supprimer un composant

L'`delete-component`exemple suivant supprime une version de construction d'un composant en spécifiant son ARN.

```
aws imagebuilder delete-component \
    --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous DeleteComponentà la section Référence des AWS CLI commandes.

`delete-image-pipeline`

L'exemple de code suivant montre comment utiliser`delete-image-pipeline`.

AWS CLI

Pour supprimer un pipeline d'images

L'`delete-image-pipeline`exemple suivant supprime un pipeline d'images en spécifiant son ARN.

```
aws imagebuilder delete-image-pipeline \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
my-example-pipeline
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous DeleteImagePipelineà la section Référence des AWS CLI commandes.

`delete-image-recipe`

L'exemple de code suivant montre comment utiliser`delete-image-recipe`.

AWS CLI

Pour supprimer une recette imagée

L'`delete-image-recipe`exemple suivant supprime une recette d'image en spécifiant son ARN.

```
aws imagebuilder delete-image-recipe \
    --image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
mybasicrecipe/2019.12.03
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous DeleteImageRecipeà la section Référence des AWS CLI commandes.

`delete-image`

L'exemple de code suivant montre comment utiliser`delete-image`.

AWS CLI

Pour supprimer une image

L'`delete-image`exemple suivant supprime une version de génération d'image en spécifiant son ARN.

```
aws imagebuilder delete-image \
    --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-
example-image/2019.12.02/1
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous DeleteImageà la section Référence des AWS CLI commandes.

`delete-infrastructure-configuration`

L'exemple de code suivant montre comment utiliser`delete-infrastructure-configuration`.

AWS CLI

Pour supprimer une configuration d'infrastructure

L'`delete-infrastructure-configuration`exemple suivant supprime un pipeline d'images en spécifiant son ARN.

```
aws imagebuilder delete-infrastructure-configuration \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous DeleteInfrastructureConfigurationà la section Référence des AWS CLI commandes.

`get-component-policy`

L'exemple de code suivant montre comment utiliser`get-component-policy`.

AWS CLI

Pour obtenir les détails de la politique relative aux composants

L'`get-component-policy`exemple suivant répertorie les détails d'une politique de composant en spécifiant son ARN.

```
aws imagebuilder get-component-policy \
    --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1
```

Sortie :

```
{
    "Policy": "{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ], "Resource":
 [ "arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1" ] } ] }"
}
```

Pour plus d'informations, consultez Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI < https://docs.aws.amazon.com/imagebuilder/ latest/userguide/ managing - image-builder-cli .html>`__ dans le guide de l'utilisateur d'EC2 Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetComponentPolicyà la section Référence des AWS CLI commandes.

`get-component`

L'exemple de code suivant montre comment utiliser`get-component`.

AWS CLI

Pour obtenir des informations détaillées sur les composants

L'`get-component`exemple suivant répertorie les détails d'un composant en spécifiant son ARN.

```
aws imagebuilder get-component \
    --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/component-name/1.0.0/1
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "component": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0/1",
        "name": "component-name",
        "version": "1.0.0",
        "type": "TEST",
        "platform": "Linux",
        "owner": "123456789012",
        "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
 testing document.\nschemaVersion: 1.0\n\nphases:\n  - name: test\n    steps:\n
   - name: HelloWorldStep\n        action: ExecuteBash\n        inputs:\n
 commands:\n            - echo \"Hello World! Test.\"\n",
        "encrypted": true,
        "dateCreated": "2020-01-27T20:43:30.306Z",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetComponentà la section Référence des AWS CLI commandes.

`get-distribution-configuration`

L'exemple de code suivant montre comment utiliser`get-distribution-configuration`.

AWS CLI

Pour obtenir les détails d'une configuration de distribution

L'`get-distribution-configuration`exemple suivant affiche les détails d'une configuration de distribution en spécifiant son ARN.

```
aws imagebuilder get-distribution-configuration \
    --distribution-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "distributionConfiguration": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/myexampledistribution",
        "name": "MyExampleDistribution",
        "description": "Copies AMI to eu-west-1 and exports to S3",
        "distributions": [
            {
                "region": "us-west-2",
                "amiDistributionConfiguration": {
                    "name": "Name {{imagebuilder:buildDate}}",
                    "description": "An example image name with parameter
 references",
                    "amiTags": {
                        "KeyName": "{{ssm:parameter_name}}"
                    },
                    "launchPermission": {
                        "userIds": [
                            "123456789012"
                        ]
                    }
                }
            },
            {
                "region": "eu-west-1",
                "amiDistributionConfiguration": {
                    "name": "My {{imagebuilder:buildVersion}} image
 {{imagebuilder:buildDate}}",
                    "amiTags": {
                        "KeyName": "Value"
                    },
                    "launchPermission": {
                        "userIds": [
                            "123456789012"
                        ]
                    }
```

```
                }
            }
        ],
        "dateCreated": "2020-02-19T18:40:10.529Z",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetDistributionConfigurationà la section Référence des AWS CLI commandes.

`get-image-pipeline`

L'exemple de code suivant montre comment utiliser`get-image-pipeline`.

AWS CLI

Pour obtenir les détails du pipeline d'images

L'`get-image-pipeline`exemple suivant répertorie les détails d'un pipeline d'images en spécifiant son ARN.

```
aws imagebuilder get-image-pipeline \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipeline": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
        "name": "MyWindows2016Pipeline",
        "description": "Builds Windows 2016 Images",
        "platform": "Windows",
        "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
        "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
```

```
        "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
        "imageTestsConfiguration": {
            "imageTestsEnabled": true,
            "timeoutMinutes": 60
        },
        "schedule": {
            "scheduleExpression": "cron(0 0 * * SUN)",
            "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
        },
        "status": "ENABLED",
        "dateCreated": "2020-02-19T19:04:01.253Z",
        "dateUpdated": "2020-02-19T19:04:01.253Z",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetImagePipelineà la section Référence des AWS CLI commandes.

`get-image-policy`

L'exemple de code suivant montre comment utiliser`get-image-policy`.

AWS CLI

Pour obtenir des informations sur la politique en matière d'images

L'`get-image-policy`exemple suivant répertorie les détails d'une politique d'image en spécifiant son ARN.

```
aws imagebuilder get-image-policy \
    --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image/2019.12.03/1
```

Sortie :

```
{
```

```
    "Policy": "{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage",
 "imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.03/1" ] } ] }"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetImagePolicyà la section Référence des AWS CLI commandes.

`get-image-recipe-policy`

L'exemple de code suivant montre comment utiliser`get-image-recipe-policy`.

AWS CLI

Pour obtenir les détails de la politique en matière de recettes illustrées

L'`get-image-recipe-policy`exemple suivant répertorie les détails d'une politique de recette d'image en spécifiant son ARN.

```
aws imagebuilder get-image-recipe-policy \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-recipe/2019.12.03/1
```

Sortie :

```
{
    "Policy": "{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":
 [ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-
recipe/2019.12.03/1" ] } ] }"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetImageRecipePolicyà la section Référence des AWS CLI commandes.

`get-image`

L'exemple de code suivant montre comment utiliser`get-image`.

AWS CLI

Pour obtenir les détails de l'image

L'`get-image`exemple suivant répertorie les détails d'une image en spécifiant son ARN.

```
aws imagebuilder get-image \
    --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "image": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1",
        "name": "MyBasicRecipe",
        "version": "2019.12.03/1",
        "platform": "Windows",
        "state": {
            "status": "BUILDING"
        },
        "imageRecipe": {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
            "name": "MyBasicRecipe",
            "description": "This example image recipe creates a Windows 2016
 image.",
            "platform": "Windows",
            "version": "2019.12.03",
            "components": [
                {
                    "componentArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
                },
                {
                    "componentArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myimportedcomponent/1.0.0/1"
                }
```

```
            ],
            "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-
server-2016-english-full-base-x86/2019.12.17/1",
            "dateCreated": "2020-02-14T19:46:16.904Z",
            "tags": {}
        },
        "infrastructureConfiguration": {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
            "name": "MyExampleInfrastructure",
            "description": "An example that will retain instances of failed builds",
            "instanceTypes": [
                "m5.large",
                "m5.xlarge"
            ],
            "instanceProfileName": "EC2InstanceProfileForImageFactory",
            "securityGroupIds": [
                "sg-a1b2c3d4"
            ],
            "subnetId": "subnet-a1b2c3d4",
            "logging": {
                "s3Logs": {
                    "s3BucketName": "bucket-name",
                    "s3KeyPrefix": "bucket-path"
                }
            },
            "keyPair": "Sam",
            "terminateInstanceOnFailure": false,
            "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
            "dateCreated": "2020-02-14T21:21:05.098Z",
            "tags": {}
        },
        "imageTestsConfiguration": {
            "imageTestsEnabled": true,
            "timeoutMinutes": 720
        },
        "dateCreated": "2020-02-14T23:14:13.597Z",
        "outputResources": {
            "amis": []
        },
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetImageà la section Référence des AWS CLI commandes.

`get-infrastructure-configuration`

L'exemple de code suivant montre comment utiliser`get-infrastructure-configuration`.

AWS CLI

Pour obtenir les détails de configuration de l'infrastructure

L'`get-infrastructure-configuration`exemple suivant répertorie les détails d'une configuration d'infrastructure en spécifiant son ARN.

```
aws imagebuilder get-infrastructure-configuration \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfiguration": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
        "name": "MyExampleInfrastructure",
        "description": "An example that will retain instances of failed builds",
        "instanceTypes": [
            "m5.large",
            "m5.xlarge"
        ],
        "instanceProfileName": "EC2InstanceProfileForImageBuilder",
        "securityGroupIds": [
            "sg-a48c95ef"
        ],
        "subnetId": "subnet-a48c95ef",
        "logging": {
            "s3Logs": {
                "s3BucketName": "bucket-name",
```

```
                "s3KeyPrefix": "bucket-path"
            }
        },
        "keyPair": "Name",
        "terminateInstanceOnFailure": false,
        "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
        "dateCreated": "2020-02-19T19:11:51.858Z",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous GetInfrastructureConfigurationà la section Référence des AWS CLI commandes.

`import-component`

L'exemple de code suivant montre comment utiliser`import-component`.

AWS CLI

Pour importer un composant

L'`import-component`exemple suivant importe un script préexistant à l'aide d'un fichier JSON.

```
aws imagebuilder import-component \
    --cli-input-json file://import-component.json
```

Contenu de `import-component.json` :

```
{
    "name": "MyImportedComponent",
    "semanticVersion": "1.0.0",
    "description": "An example of how to import a component",
    "changeDescription": "First commit message.",
    "format": "SHELL",
    "platform": "Windows",
    "type": "BUILD",
    "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myimportedcomponent/1.0.0/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous ImportComponentà la section Référence des AWS CLI commandes.

`list-component-build-versions`

L'exemple de code suivant montre comment utiliser`list-component-build-versions`.

AWS CLI

Pour répertorier les versions de construction des composants

L'`list-component-build-versions`exemple suivant répertorie les versions de construction des composants avec une version sémantique spécifique.

```
aws imagebuilder list-component-build-versions --component-
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1",
            "name": "MyExampleComponent",
            "version": "2019.12.02",
            "platform": "Windows",
            "type": "BUILD",
```

```
            "owner": "123456789012",
            "description": "An example component that builds, validates and tests an
 image",
            "changeDescription": "Initial version.",
            "dateCreated": "2020-02-19T18:53:45.940Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous ListComponentBuildVersionsà la section Référence des AWS CLI commandes.

`list-components`

L'exemple de code suivant montre comment utiliser`list-components`.

AWS CLI

Pour répertorier toutes les versions sémantiques des composants

L'`list-components`exemple suivant répertorie toutes les versions sémantiques des composants auxquelles vous avez accès. Vous pouvez éventuellement filtrer les composants qui vous appartiennent, ceux d'Amazon ou ceux qui ont été partagés avec vous par d'autres comptes.

```
aws imagebuilder list-components
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentVersionList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0",
            "name": "component-name",
            "version": "1.0.0",
```

```
            "platform": "Linux",
            "type": "TEST",
            "owner": "123456789012",
            "dateCreated": "2020-01-27T20:43:30.306Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, reportez-vous ListComponentsà la section Référence des AWS CLI commandes.

`list-distribution-configurations`

L'exemple de code suivant montre comment utiliser`list-distribution-configurations`.

AWS CLI

Pour répertorier les distributions

L'`list-distribution-configurations`exemple suivant répertorie toutes vos distributions.

```
aws imagebuilder list-distribution-configurations
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "distributionConfigurationSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/myexampledistribution",
            "name": "MyExampleDistribution",
            "description": "Copies AMI to eu-west-1 and exports to S3",
            "dateCreated": "2020-02-19T18:40:10.529Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListDistributionConfigurationsla section Référence des AWS CLI commandes.

`list-image-build-versions`

L'exemple de code suivant montre comment utiliser`list-image-build-versions`.

AWS CLI

Pour répertorier les versions de génération d'images

L'`list-image-build-versions`exemple suivant répertorie toutes les versions de génération d'image avec une version sémantique.

```
aws imagebuilder list-image-build-versions \
    --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/7",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/7",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "Can't start SSM Automation for arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/7 during
 building. Parameter \"iamInstanceProfileName\" has a null value."
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-19T18:56:11.511Z",
            "outputResources": {
```

```
                "amis": []
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/6",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/6",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "An internal error has occurred."
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T22:49:08.142Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T22-49-38.704Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    },
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "Name 2020-02-18T22-49-08.131Z",
                        "description": "Copies AMI to eu-west-2 and exports to S3"
                    },
                    {
                        "region": "eu-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "My 6 image 2020-02-18T22-49-08.131Z",
                        "description": "Copies AMI to eu-west-2 and exports to S3"
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/5",
```

```
            "name": "MyBasicRecipe",
            "version": "2019.12.03/5",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T16:51:48.403Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T16-52-18.965Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/4",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/4",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T16:50:01.827Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T16-50-32.280Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
```

```
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/3",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/3",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-14T23:14:13.597Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-14T23-14-44.243Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/2",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/2",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "SSM execution 'a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'
 failed with status = 'Failed' and failure message = 'Step fails when it is
 verifying the command has completed. Command a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 returns unexpected invocation result: \n{Status=[Failed], ResponseCode=[1],
 Output=[\n----------ERROR-------\nfailed to run commands: exit status 1],
 OutputPayload=[{\"Status\":\"Failed\",\"ResponseCode\":1,\"Output\":\"\
\n----------ERROR-------\\nfailed to run commands: exit status 1\",\"CommandId\":
\"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111\"}], CommandId=[a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111]}. Please refer to Automation Service Troubleshooting Guide for more
 diagnosis details.'"
            },
```

```
            "owner": "123456789012",
            "dateCreated": "2020-02-14T22:57:42.593Z",
            "outputResources": {
                "amis": []
            },
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListImageBuildVersionsla section Référence des AWS CLI commandes.

`list-image-pipeline-images`

L'exemple de code suivant montre comment utiliser`list-image-pipeline-images`.

AWS CLI

Pour répertorier les images du pipeline d'images

L'`list-image-pipeline-images`exemple suivant répertorie toutes les images créées par un pipeline d'images spécifique.

```
aws imagebuilder list-image-pipeline-images \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipelineList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
            "name": "MyWindows2016Pipeline",
            "description": "Builds Windows 2016 Images",
            "platform": "Windows",
```

```
            "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-
recipe/mybasicrecipe/2019.12.03",
            "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
            "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
            "imageTestsConfiguration": {
                "imageTestsEnabled": true,
                "timeoutMinutes": 60
            },
            "schedule": {
                "scheduleExpression": "cron(0 0 * * SUN)",
                "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
            },
            "status": "ENABLED",
            "dateCreated": "2020-02-19T19:04:01.253Z",
            "dateUpdated": "2020-02-19T19:04:01.253Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/sam",
            "name": "PipelineName",
            "platform": "Linux",
            "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-
recipe/recipe-name-a1b2c3d45678/1.0.0",
            "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/infrastructureconfiguration-name-
a1b2c3d45678",
            "imageTestsConfiguration": {
                "imageTestsEnabled": true,
                "timeoutMinutes": 720
            },
            "status": "ENABLED",
            "dateCreated": "2019-12-16T18:19:02.068Z",
            "dateUpdated": "2019-12-16T18:19:02.068Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListImagePipelineImagesla section Référence des AWS CLI commandes.

`list-image-recipes`

L'exemple de code suivant montre comment utiliser`list-image-recipes`.

AWS CLI

Pour répertorier les recettes d'images

L'`list-image-recipes`exemple suivant répertorie toutes vos recettes d'images.

```
aws imagebuilder list-image-recipes
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
            "name": "MyBasicRecipe",
            "platform": "Windows",
            "owner": "123456789012",
            "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-
server-2016-english-full-base-x86/2019.x.x",
            "dateCreated": "2020-02-19T18:54:25.975Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/recipe-
name-a1b2c3d45678/1.0.0",
            "name": "recipe-name-a1b2c3d45678",
            "platform": "Linux",
```

```
            "owner": "123456789012",
            "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/amazon-linux-2-
x86/2019.11.21",
            "dateCreated": "2019-12-16T18:19:00.120Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListImageRecipesla section Référence des AWS CLI commandes.

`list-images`

L'exemple de code suivant montre comment utiliser`list-images`.

AWS CLI

Pour répertorier des images

L'`list-images`exemple suivant répertorie toutes les versions sémantiques auxquelles vous avez accès.

```
aws imagebuilder list-images
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageVersionList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03",
            "name": "MyBasicRecipe",
            "version": "2019.12.03",
            "platform": "Windows",
            "owner": "123456789012",
            "dateCreated": "2020-02-14T21:29:18.810Z"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListImagesla section Référence des AWS CLI commandes.

`list-infrastructure-configurations`

L'exemple de code suivant montre comment utiliser`list-infrastructure-configurations`.

AWS CLI

Pour répertorier les configurations d'infrastructure

L'`list-infrastructure-configurations`exemple suivant répertorie toutes les configurations de votre infrastructure.

```
aws imagebuilder list-infrastructure-configurations
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfigurationSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
            "name": "MyExampleInfrastructure",
            "description": "An example that will retain instances of failed builds",
            "dateCreated": "2020-02-19T19:11:51.858Z",
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/infrastructureconfiguration-name-a1b2c3d45678",
            "name": "infrastructureConfiguration-name-a1b2c3d45678",
            "dateCreated": "2019-12-16T18:19:01.038Z",
            "tags": {
                "KeyName": "KeyValue"
            }
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListInfrastructureConfigurationsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource spécifique

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'une ressource spécifique.

```
aws imagebuilder list-tags-for-resource \
    --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

Sortie :

```
{
    "tags": {
        "KeyName": "KeyValue"
    }
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`put-component-policy`

L'exemple de code suivant montre comment utiliser`put-component-policy`.

AWS CLI

Pour appliquer une politique de ressources à un composant

La `put-component-policy` commande suivante applique une politique de ressources à un composant de build afin de permettre le partage entre comptes des composants de build. Nous vous recommandons d'utiliser la commande RAM CLI`create-resource-share`. Si vous utilisez la commande EC2 Image Builder CLI`put-component-policy`, vous devez également utiliser la commande `promote-resource-share-create-from-policy` RAM CLI pour que la ressource soit visible par tous les principaux avec lesquels la ressource est partagée.

```
aws imagebuilder put-component-policy \
    --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1 \
    --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ],
 "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1" ] } ] }'
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir PutComponentPolicyla section Référence des AWS CLI commandes.

`put-image-policy`

L'exemple de code suivant montre comment utiliser`put-image-policy`.

AWS CLI

Pour appliquer une politique de ressources à une image

La `put-image-policy` commande suivante applique une politique de ressources à une image afin de permettre le partage d'images entre comptes. Nous vous recommandons d'utiliser la commande RAM CLI create-resource-share. Si vous utilisez la commande EC2 Image Builder CLI put-image-policy, vous devez également utiliser la commande RAM CLI promote-resource-share-create -from-policy afin que la ressource soit visible par tous les principaux avec lesquels la ressource est partagée.

```
aws imagebuilder put-image-policy \
    --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1 \
    --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage", "imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1" ] } ] }'
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir PutImagePolicyla section Référence des AWS CLI commandes.

`put-image-recipe-policy`

L'exemple de code suivant montre comment utiliser`put-image-recipe-policy`.

AWS CLI

Pour appliquer une politique de ressources à une recette d'image

La `put-image-recipe-policy` commande suivante applique une politique de ressources à une recette d'image afin de permettre le partage de recettes d'images entre comptes. Nous vous recommandons d'utiliser la commande RAM CLI`create-resource-share`. Si vous utilisez

la commande EC2 Image Builder CLIput-image-recipe-policy, vous devez également utiliser la commande promote-resource-share-create-from-policy RAM CLI pour que la ressource soit visible par tous les principaux avec lesquels la ressource est partagée.

```
aws imagebuilder put-image-recipe-policy \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
example-image-recipe/2019.12.02 \
    --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":
 [ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/example-image-
recipe/2019.12.02" ] } ] }'
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
example-image-recipe/2019.12.02/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir PutImageRecipePolicyla section Référence des AWS CLI commandes.

`start-image-pipeline-execution`

L'exemple de code suivant montre comment utiliserstart-image-pipeline-execution.

AWS CLI

Pour démarrer un pipeline d'images manuellement

L'start-image-pipeline-executionexemple suivant démarre manuellement un pipeline d'images.

```
aws imagebuilder start-image-pipeline-execution \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir StartImagePipelineExecutionla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

L'`tag-resource`exemple suivant ajoute et balise une ressource dans EC2 Image Builder à l'aide d'un fichier JSON.

```
aws imagebuilder tag-resource \
    --cli-input-json file://tag-resource.json
```

Contenu de `tag-resource.json` :

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "tags": {
        "KeyName: "KeyValue"
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime une balise d'une ressource à l'aide d'un fichier JSON.

```
aws imagebuilder untag-resource \
    --cli-input-json file://tag-resource.json
```

Contenu de `untag-resource.json` :

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "tagKeys": [
        "KeyName"
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-distribution-configuration`

L'exemple de code suivant montre comment utiliser`update-distribution-configuration`.

AWS CLI

Pour mettre à jour une configuration de distribution

L'`update-distribution-configuration`exemple suivant met à jour une configuration de distribution à l'aide d'un fichier JSON.

```
aws imagebuilder update-distribution-configuration \
    --cli-input-json file://update-distribution-configuration.json
```

Contenu de `update-distribution-configuration.json` :

```
{
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
    "description": "Copies AMI to eu-west-2 and exports to S3",
    "distributions": [
        {
            "region": "us-west-2",
            "amiDistributionConfiguration": {
                "name": "Name {{imagebuilder:buildDate}}",
                "description": "An example image name with parameter references"
            }
        },
        {
            "region": "eu-west-2",
            "amiDistributionConfiguration": {
                "name": "My {{imagebuilder:buildVersion}} image
 {{imagebuilder:buildDate}}"
            }
        }
    ]
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir UpdateDistributionConfigurationla section Référence des AWS CLI commandes.

`update-image-pipeline`

L'exemple de code suivant montre comment utiliser`update-image-pipeline`.

AWS CLI

Pour mettre à jour un pipeline d'images

L'`update-image-pipeline`exemple suivant met à jour un pipeline d'images à l'aide d'un fichier JSON.

```
aws imagebuilder update-image-pipeline \
    --cli-input-json file://update-image-pipeline.json
```

Contenu de `update-image-pipeline.json` :

```
{
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
    "imageTestsConfiguration": {
        "imageTestsEnabled": true,
        "timeoutMinutes": 120
    },
    "schedule": {
        "scheduleExpression": "cron(0 0 * * MON)",
        "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "DISABLED"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir UpdateImagePipelinela section Référence des AWS CLI commandes.

`update-infrastructure-configuration`

L'exemple de code suivant montre comment utiliser`update-infrastructure-configuration`.

AWS CLI

Pour mettre à jour la configuration d'une infrastructure

L'`update-infrastructure-configuration`exemple suivant met à jour une configuration d'infrastructure à l'aide d'un fichier JSON.

```
aws imagebuilder update-infrastructure-configuration \
    --cli-input-json file:/update-infrastructure-configuration.json
```

Contenu de `update-infrastructure-configuration.json` :

```
{
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "description": "An example that will terminate instances of failed builds",
    "instanceTypes": [
        "m5.large", "m5.2xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageFactory",
    "securityGroupIds": [
        "sg-a48c95ef"
    ],
    "subnetId": "subnet-a48c95ef",
    "logging": {
        "s3Logs": {
            "s3BucketName": "bucket-name",
```

```
            "s3KeyPrefix": "bucket-path"
        }
    },
    "terminateInstanceOnFailure": true,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name"
}
```

Sortie :

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Configuration et gestion d'un pipeline d' EC2 images Image Builder à l'aide de la AWS CLI dans le guide de l'utilisateur EC2 d'Image Builder.

- Pour plus de détails sur l'API, voir UpdateInfrastructureConfigurationla section Référence des AWS CLI commandes.

Exemples d'utilisation d'Incident Manager AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Incident Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-replication-set`

L'exemple de code suivant montre comment utiliser`create-replication-set`.

AWS CLI

Pour créer le jeu de réplication

L'`create-replication-set`exemple suivant crée le jeu de réplication utilisé par Incident Manager pour répliquer et chiffrer les données de votre compte Amazon Web Services. Cet exemple utilise les régions us-east-1 et us-east-2 lors de la création du jeu de réplication.

```
aws ssm-incidents create-replication-set \
    --regions '{"us-east-1": {"sseKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}, "us-east-2":
 {"sseKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}}'
```

Sortie :

```
{
    "replicationSetArns": [
        "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du jeu de réplication Incident Manager dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous CreateReplicationSetà la section Référence des AWS CLI commandes.

`create-response-plan`

L'exemple de code suivant montre comment utiliser`create-response-plan`.

AWS CLI

Pour créer un plan d'intervention

L'`create-response-plan`exemple suivant crée un plan de réponse avec les détails spécifiés.

```
aws ssm-incidents create-response-plan \
    --chat-channel '{"chatbotSns": ["arn:aws:sns:us-
east-1:111122223333:Standard_User"]}' \
```

```
    --display-name "Example response plan" \
    --incident-template '{"impact": 5, "title": "example-incident"}' \
    --name "example-response" \
    --actions '[{"ssmAutomation": {"documentName": "AWSIncidents-
CriticalIncidentRunbookTemplate", "documentVersion": "$DEFAULT",
 "roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager", "targetAccount":
 "RESPONSE_PLAN_OWNER_ACCOUNT"}}]' \
    --engagements '["arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"]'
```

Sortie :

```
{
    "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}
```

Pour plus d'informations, consultez la section Préparation des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous CreateResponsePlanà la section Référence des AWS CLI commandes.

`create-timeline-event`

L'exemple de code suivant montre comment utiliser`create-timeline-event`.

AWS CLI

Exemple 1 : pour créer un événement chronologique personnalisé

L'`create-timeline-event`exemple suivant crée un événement chronologique personnalisé à l'heure spécifiée lors de l'incident spécifié.

```
aws ssm-incidents create-timeline-event \
    --event-data "\"example timeline event\"" \
    --event-time 2022-10-01T20:30:00.000 \
    --event-type "Custom Event" \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
```

Sortie :

```
{
    "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

Exemple 2 : pour créer un événement chronologique avec une note d'incident

L'`create-timeline-event`exemple suivant crée un événement chronologique répertorié dans le panneau « Notes d'incident ».

```
aws ssm-incidents create-timeline-event \
     --event-data "\"New Note\"" \
     --event-type "Note" \
     --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE" \
     --event-time 2023-06-20T12:06:00.000 \
     --event-references '[{"resource":"arn:aws:ssm-incidents::111122223333:incident-
record/Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE"}]'
```

Sortie :

```
{
    "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous CreateTimelineEventà la section Référence des AWS CLI commandes.

`delete-incident-record`

L'exemple de code suivant montre comment utiliser`delete-incident-record`.

AWS CLI

Pour supprimer un enregistrement d'incident

L'`delete-incident-record`exemple suivant supprime l'enregistrement d'incident spécifié.

```
aws ssm-incidents delete-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suivi des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteIncidentRecordà la section Référence des AWS CLI commandes.

`delete-replication-set`

L'exemple de code suivant montre comment utiliser`delete-replication-set`.

AWS CLI

Pour supprimer le jeu de réplication

L'`delete-replication-set`exemple suivant supprime le jeu de réplication de votre compte Amazon Web Services. La suppression du jeu de réplication entraîne également la suppression de toutes les données d'Incident Manager. Cela ne peut pas être annulé.

```
aws ssm-incidents delete-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-bb3f-413c-08df53673b57"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du jeu de réplication Incident Manager dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteReplicationSetà la section Référence des AWS CLI commandes.

`delete-resource-policy`

L'exemple de code suivant montre comment utiliser`delete-resource-policy`.

AWS CLI

Pour supprimer une politique de ressources

L'`delete-resource-policy`exemple suivant supprime une politique de ressources d'un plan de réponse. Cela révoquera l'accès du directeur ou de l'organisation avec lequel le plan d'intervention a été partagé.

```
aws ssm-incidents delete-resource-policy \
    --policy-id "be8b57191f0371f1c6827341aa3f0a03" \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de contacts partagés et de plans de réponse dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteResourcePolicyà la section Référence des AWS CLI commandes.

`delete-response-plan`

L'exemple de code suivant montre comment utiliser`delete-response-plan`.

AWS CLI

Pour supprimer un plan de réponse

L'`delete-response-plan`exemple suivant supprime le plan de réponse spécifié.

```
aws ssm-incidents delete-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Préparation des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteResponsePlanà la section Référence des AWS CLI commandes.

`delete-timeline-event`

L'exemple de code suivant montre comment utiliser`delete-timeline-event`.

AWS CLI

Pour supprimer un événement chronologique

L'`delete-timeline-event`exemple suivant supprime un événement chronologique personnalisé de l'enregistrement d'incident spécifié.

```
aws ssm-incidents delete-timeline-event \
    --event-id "c0bcc885-a41d-eb01-b4ab-9d2de193643c" \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteTimelineEventà la section Référence des AWS CLI commandes.

`get-incident-record`

L'exemple de code suivant montre comment utiliser`get-incident-record`.

AWS CLI

Pour obtenir un enregistrement d'incident

L'`get-incident-record`exemple suivant permet d'obtenir des informations sur l'enregistrement d'incident spécifié.

```
aws ssm-incidents get-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Sortie :

```
{
    "incidentRecord": {
```

```
        "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-
Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",
        "automationExecutions": [],
        "creationTime": "2021-05-21T18:16:57.579000+00:00",
        "dedupeString": "c4bcc812-85e7-938d-2b78-17181176ee1a",
        "impact": 5,
        "incidentRecordSource": {
            "createdBy": "arn:aws:iam::111122223333:user/draliatp",
            "invokedBy": "arn:aws:iam::111122223333:user/draliatp",
            "source": "aws.ssm-incidents.custom"
        },
        "lastModifiedBy": "arn:aws:iam::111122223333:user/draliatp",
        "lastModifiedTime": "2021-05-21T18:16:59.149000+00:00",
        "notificationTargets": [],
        "status": "OPEN",
        "title": "Example-Incident"
    }
}
```

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetIncidentRecordà la section Référence des AWS CLI commandes.

`get-replication-set`

L'exemple de code suivant montre comment utiliser`get-replication-set`.

AWS CLI

Pour obtenir le kit de réplication

L'`get-replication-set`exemple suivant fournit les détails du jeu de réplication utilisé par Incident Manager pour répliquer et chiffrer les données de votre compte Amazon Web Services.

```
aws ssm-incidents get-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
```

Sortie :

```
{
```

```
    "replicationSet": {
        "createdBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
        "createdTime": "2021-05-14T17:57:22.010000+00:00",
        "deletionProtected": false,
        "lastModifiedBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
        "lastModifiedTime": "2021-05-14T17:57:22.010000+00:00",
        "regionMap": {
            "us-east-1": {
                "sseKmsKeyId": "DefaultKey",
                "status": "ACTIVE"
            },
            "us-east-2": {
                "sseKmsKeyId": "DefaultKey",
                "status": "ACTIVE",
                "statusMessage": "Tagging inaccessible"
            }
        },
        "status": "ACTIVE"
    }
}
```

Pour plus d'informations, consultez la section Utilisation du jeu de réplication Incident Manager dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetReplicationSetà la section Référence des AWS CLI commandes.

`get-resource-policies`

L'exemple de code suivant montre comment utiliser`get-resource-policies`.

AWS CLI

Pour répertorier les politiques de ressources pour un plan de réponse

L'`command-name`exemple suivant répertorie les politiques de ressources associées au plan de réponse spécifié.

```
aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Sortie :

```
{
    "resourcePolicies": [
        {
            "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid
\":\"d901b37a-dbb0-458a-8842-75575c464219-external-principals\",\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::222233334444:root\"},\"Action
\":[\"ssm-incidents:GetResponsePlan\",\"ssm-incidents:StartIncident\",\"ssm-
incidents:UpdateIncidentRecord\",\"ssm-incidents:GetIncidentRecord\",\"ssm-
incidents:CreateTimelineEvent\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-
incidents:GetTimelineEvent\",\"ssm-incidents:ListTimelineEvents\",\"ssm-
incidents:UpdateRelatedItems\",\"ssm-incidents:ListRelatedItems\"],\"Resource\":
[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\",
\"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*
\"]}]}",
            "policyId": "be8b57191f0371f1c6827341aa3f0a03",
            "ramResourceShareRegion": "us-east-1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de contacts partagés et de plans de réponse dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetResourcePoliciesà la section Référence des AWS CLI commandes.

`get-response-plan`

L'exemple de code suivant montre comment utiliser`get-response-plan`.

AWS CLI

Pour obtenir les détails d'un plan d'intervention

L'`command-name`exemple suivant permet d'obtenir des informations sur un plan de réponse spécifié dans votre AWS compte.

```
aws ssm-incidents get-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Sortie :

```
{
    "actions": [
        {
            "ssmAutomation": {
                "documentName": "AWSIncidents-CriticalIncidentRunbookTemplate",
                "documentVersion": "$DEFAULT",
                "roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager",
                "targetAccount": "RESPONSE_PLAN_OWNER_ACCOUNT"
            }
        }
    ],
    "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-
Plan",
    "chatChannel": {
        "chatbotSns": [
            "arn:aws:sns:us-east-1:111122223333:Standard_User"
        ]
    },
    "displayName": "Example response plan",
    "engagements": [
        "arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"
    ],
    "incidentTemplate": {
        "impact": 5,
        "title": "Example-Incident"
    },
    "name": "Example-Response-Plan"
}
```

Pour plus d'informations, consultez la section Préparation des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetResponsePlanà la section Référence des AWS CLI commandes.

`get-timeline-event`

L'exemple de code suivant montre comment utiliser`get-timeline-event`.

AWS CLI

Pour obtenir les détails d'un événement chronologique

L'`get-timeline-event`exemple suivant renvoie les détails de l'événement chronologique spécifié.

```
aws ssm-incidents get-timeline-event \
    --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Sortie :

```
{
    "event": {
        "eventData": "\"Incident Started\"",
        "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",
        "eventTime": "2021-05-21T18:16:57+00:00",
        "eventType": "Custom Event",
        "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
        "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
    }
}
```

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetTimelineEventà la section Référence des AWS CLI commandes.

`list-incident-records`

L'exemple de code suivant montre comment utiliser`list-incident-records`.

AWS CLI

Pour répertorier les enregistrements d'incidents

L'`command-name`exemple suivant répertorie les enregistrements d'incidents de votre compte Amazon Web Services.

```
aws ssm-incidents list-incident-records
```

Sortie :

```
{
    "incidentRecordSummaries": [
        {
            "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-
Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",
            "creationTime": "2021-05-21T18:16:57.579000+00:00",
            "impact": 5,
            "incidentRecordSource": {
                "createdBy": "arn:aws:iam::111122223333:user/draliatp",
                "invokedBy": "arn:aws:iam::111122223333:user/draliatp",
                "source": "aws.ssm-incidents.custom"
            },
            "status": "OPEN",
            "title": "Example-Incident"
        }
    ]
}
```

Pour plus d'informations, consultez la liste des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListIncidentRecordsà la section Référence des AWS CLI commandes.

`list-related-items`

L'exemple de code suivant montre comment utiliser`list-related-items`.

AWS CLI

Pour répertorier des articles connexes

L'`list-related-items`exemple suivant répertorie les éléments associés à l'incident spécifié.

```
aws ssm-incidents list-related-items \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Sortie :

```
{
    "relatedItems": [
```

```
        {
            "identifier": {
                "type": "OTHER",
                "value": {
                    "url": "https://console.aws.amazon.com/systems-manager/opsitems/
oi-8ef82158e190/workbench?region=us-east-1"
                }
            },
            "title": "Example related item"
        },
        {
            "identifier": {
                "type": "PARENT",
                "value": {
                    "arn": "arn:aws:ssm:us-east-1:111122223333:opsitem/
oi-8084126392ac"
                }
            },
            "title": "parentItem"
        }
    ]
}
```

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListRelatedItemsà la section Référence des AWS CLI commandes.

`list-replication-sets`

L'exemple de code suivant montre comment utiliser`list-replication-sets`.

AWS CLI

Pour répertorier le jeu de réplication

L'`list-replication-set`exemple suivant répertorie le jeu de réplication utilisé par Incident Manager pour répliquer et chiffrer les données de votre AWS compte.

```
aws ssm-incidents list-replication-sets
```

Sortie :

```
{
    "replicationSetArns": [
        "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
    ]
}
```

Pour plus d'informations, consultez la section Utilisation du jeu de réplication Incident Manager dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListReplicationSetsà la section Référence des AWS CLI commandes.

`list-response-plans`

L'exemple de code suivant montre comment utiliser`list-response-plans`.

AWS CLI

Pour répertorier les plans de réponse disponibles

L'`list-response-plans`exemple suivant répertorie les plans de réponse disponibles dans votre compte Amazon Web Services.

```
aws ssm-incidents list-response-plans
```

Sortie :

```
{
    "responsePlanSummaries": [
        {
            "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan",
            "displayName": "Example response plan",
            "name": "Example-Response-Plan"
        }
    ]
}
```

Pour plus d'informations, consultez la section Préparation des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListResponsePlansà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un plan de réponse

L'`list-tags-for-resource`exemple suivant répertorie les balises associées au plan de réponse spécifié.

```
aws ssm-incidents list-tags-for-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Sortie :

```
{
    "tags": {
    "group1": "1"
    }
}
```

Pour plus d'informations, consultez la section Balisage dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-timeline-events`

L'exemple de code suivant montre comment utiliser`list-timeline-events`.

AWS CLI

Pour répertorier les événements chronologiques d'un incident

L'`command-name`exemple suivant répertorie les événements chronologiques de l'incident spécifié.

```
aws ssm-incidents list-timeline-events \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Sortie :

```
{
    "eventSummaries": [
        {
            "eventId": "8cbcc889-35e1-a42d-2429-d6f100799915",
            "eventTime": "2021-05-21T22:36:13.766000+00:00",
            "eventType": "SSM Incident Record Update",
            "eventUpdatedTime": "2021-05-21T22:36:13.766000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "a2bcc825-aab5-1787-c605-f9bb2640d85b",
            "eventTime": "2021-05-21T18:58:46.443000+00:00",
            "eventType": "SSM Incident Record Update",
            "eventUpdatedTime": "2021-05-21T18:58:46.443000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "5abcc812-89c0-b0a8-9437-1c74223d4685",
            "eventTime": "2021-05-21T18:16:59.149000+00:00",
            "eventType": "SSM Incident Record Update",
            "eventUpdatedTime": "2021-05-21T18:16:59.149000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "06bcc812-8820-405e-4065-8d2b14d29b92",
            "eventTime": "2021-05-21T18:16:58+00:00",
            "eventType": "SSM Automation Execution Start Failure for Incident",
            "eventUpdatedTime": "2021-05-21T18:16:58.689000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",
            "eventTime": "2021-05-21T18:16:57+00:00",
```

```
            "eventType": "Custom Event",
            "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "c0bcc885-a41d-eb01-b4ab-9d2de193643c",
            "eventTime": "2020-10-01T20:30:00+00:00",
            "eventType": "Custom Event",
            "eventUpdatedTime": "2021-05-21T22:28:26.299000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        }
    ]
}
```

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListTimelineEventsà la section Référence des AWS CLI commandes.

`put-resource-policy`

L'exemple de code suivant montre comment utiliser`put-resource-policy`.

AWS CLI

Pour partager un plan de réponse et des incidents

L'`command-name`exemple suivant ajoute une politique de ressources au Example-Response-Plan qui partage le plan de réponse et les incidents associés avec le principal spécifié.

```
aws ssm-incidents put-resource-policy \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan" \
    --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"ExampleResourcePolciy\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"arn:aws:iam::222233334444:root\"},\"Action\":[\"ssm-incidents:GetResponsePlan
\",\"ssm-incidents:StartIncident\",\"ssm-incidents:UpdateIncidentRecord
\",\"ssm-incidents:GetIncidentRecord\",\"ssm-incidents:CreateTimelineEvent
\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-incidents:GetTimelineEvent
\",\"ssm-incidents:ListTimelineEvents\",\"ssm-incidents:UpdateRelatedItems
```

```
\",\"ssm-incidents:ListRelatedItems\"],\"Resource\":[\"arn:aws:ssm-
incidents:*:111122223333:response-plan/Example-Response-Plan\",\"arn:aws:ssm-
incidents:*:111122223333:incident-record/Example-Response-Plan/*\"]}]}"
```

Sortie :

```
{
    "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

Pour plus d'informations, consultez la section Utilisation de contacts partagés et de plans de réponse dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous PutResourcePolicyà la section Référence des AWS CLI commandes.

`start-incident`

L'exemple de code suivant montre comment utiliser`start-incident`.

AWS CLI

Pour démarrer un incident

L'`start-incident`exemple suivant démarre un incident en utilisant le plan de réponse spécifié.

```
aws ssm-incidents start-incident \
    --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan"
```

Sortie :

```
{
    "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
}
```

Pour plus d'informations, consultez la section Création d'incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous StartIncidentà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un plan de réponse

L'`tag-resource`exemple suivant balise un plan de réponse spécifié avec la paire clé-valeur de balise fournie.

```
aws ssm-incidents tag-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \
    --tags '{"group1":"1"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'un plan de réponse

L'`untag-resource`exemple suivant supprime les balises spécifiées du plan de réponse.

```
aws ssm-incidents untag-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \
    --tag-keys '["group1"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-deletion-protection`

L'exemple de code suivant montre comment utiliser`update-deletion-protection`.

AWS CLI

Pour mettre à jour la réplication, définissez la protection contre la suppression

L'`update-deletion-protection`exemple suivant met à jour la protection contre la suppression de votre compte afin de vous empêcher de supprimer la dernière région de votre jeu de réplication.

```
aws ssm-incidents update-deletion-protection \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/
a2bcc5c9-0f53-8047-7fef-c20749989b40" \
    --deletion-protected
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du jeu de réplication Incident Manager dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateDeletionProtectionà la section Référence des AWS CLI commandes.

`update-incident-record`

L'exemple de code suivant montre comment utiliser`update-incident-record`.

AWS CLI

Pour mettre à jour un enregistrement d'incident

L'`command-name`exemple suivant résout l'incident spécifié.

```
aws ssm-incidents update-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
```

```
    --status "RESOLVED"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateIncidentRecordà la section Référence des AWS CLI commandes.

`update-related-items`

L'exemple de code suivant montre comment utiliser`update-related-items`.

AWS CLI

Pour mettre à jour un élément lié à un incident

L'`update-related-item`exemple suivant supprime un élément connexe de l'enregistrement d'incident spécifié.

```
aws ssm-incidents update-related-items \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
    --related-items-update '{"itemToRemove": {"type": "OTHER", "value": {"url":
 "https://console.aws.amazon.com/systems-manager/opsitems/oi-8ef82158e190/workbench?
region=us-east-1"}}}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateRelatedItemsà la section Référence des AWS CLI commandes.

`update-replication-set`

L'exemple de code suivant montre comment utiliser`update-replication-set`.

AWS CLI

Pour mettre à jour un jeu de réplication

L'`command-name`exemple suivant supprime la région us-east-2 du jeu de réplication.

```
aws ssm-incidents update-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/
a2bcc5c9-0f53-8047-7fef-c20749989b40" \
    --actions '[{"deleteRegionAction": {"regionName": "us-east-2"}}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation du jeu de réplication Incident Manager dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateReplicationSetà la section Référence des AWS CLI commandes.

`update-response-plan`

L'exemple de code suivant montre comment utiliser`update-response-plan`.

AWS CLI

Pour mettre à jour un plan de réponse

L'`update-response-plan`exemple suivant supprime un canal de discussion du plan de réponse spécifié.

```
aws ssm-incidents update-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
 \
    --chat-channel '{"empty":{}}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Préparation des incidents dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateResponsePlanà la section Référence des AWS CLI commandes.

`update-timeline-event`

L'exemple de code suivant montre comment utiliser`update-timeline-event`.

AWS CLI

Pour mettre à jour un événement chronologique

L'`update-timeline-event`exemple suivant met à jour l'heure à laquelle l'événement s'est produit.

```
aws ssm-incidents update-timeline-event \
    --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
    --event-time "2021-05-21T18:10:57+00:00"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les détails de l'incident dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateTimelineEventà la section Référence des AWS CLI commandes.

Exemples de contacts du gestionnaire d'incidents utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide des contacts du gestionnaire d'incidents.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-page`

L'exemple de code suivant montre comment utiliser`accept-page`.

AWS CLI

Pour accepter une page pendant un engagement

L'`accept-page`exemple suivant utilise un code d'acceptation envoyé au canal de contact pour accepter une page.

```
aws ssm-contacts accept-page \
    --page-id "arn:aws:ssm-contacts:us-east-2:682428703967:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3" \
    --accept-type READ \
    --accept-code 425440
```

Cette commande ne produit aucune sortie

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous AcceptPageà la section Référence des AWS CLI commandes.

`activate-contact-channel`

L'exemple de code suivant montre comment utiliser`activate-contact-channel`.

AWS CLI

Activer le canal de contact d'un contact

L'`activate-contact-channel`exemple suivant active un canal de contact et le rend utilisable dans le cadre d'un incident.

```
aws ssm-contacts activate-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d" \
    --activation-code "466136"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ActivateContactChannelà la section Référence des AWS CLI commandes.

`command-name`

L'exemple de code suivant montre comment utiliser`command-name`.

AWS CLI

Pour supprimer un contact

L'`command-name`exemple suivant supprime un contact. Le contact ne sera plus joignable à partir d'un plan d'escalade qui le désigne.

```
aws ssm-contacts delete-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous CommandNameà la section Référence des AWS CLI commandes.

`create-contact-channel`

L'exemple de code suivant montre comment utiliser`create-contact-channel`.

AWS CLI

Pour créer un canal de contact

Crée un canal de contact de type SMS pour le contact Akua Mansa. Les canaux de contact peuvent être créés de type SMS, EMAIL ou VOICE.

```
aws ssm-contacts create-contact-channel \
    --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --name "akuas sms-test" \
    --type SMS \
    --delivery-address '{"SimpleAddress": "+15005550199"}'
```

Sortie :

```
{
    "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous CreateContactChannelà la section Référence des AWS CLI commandes.

`create-contact`

L'exemple de code suivant montre comment utiliser`create-contact`.

AWS CLI

Pour créer un contact

L'`create-contact`exemple suivant crée un contact dans votre environnement avec un plan vierge. Le plan peut être mis à jour après avoir créé des canaux de contact. Utilisez la create-contact-channel commande avec l'ARN de sortie de cette commande. Après avoir créé des canaux de contact pour ce contact, utilisez update-contact pour mettre à jour le plan.

```
aws ssm-contacts create-contact \
    --alias "akuam" \
    --display-name "Akua Mansa" \
    --type PERSONAL \
    --plan '{"Stages": []}'
```

Sortie :

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous CreateContactà la section Référence des AWS CLI commandes.

`deactivate-contact-channel`

L'exemple de code suivant montre comment utiliser`deactivate-contact-channel`.

AWS CLI

Pour désactiver un canal de contact

L'`deactivate-contact-channel`exemple suivant désactive un canal de contact. La désactivation d'un canal de contact signifie que le canal de contact ne sera plus paginé lors d'un incident. Vous pouvez également réactiver un canal de contact à tout moment à l'aide de la `activate-contact-channel` commande.

```
aws ssm-contacts deactivate-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeactivateContactChannelà la section Référence des AWS CLI commandes.

`delete-contact-channel`

L'exemple de code suivant montre comment utiliser`delete-contact-channel`.

AWS CLI

Pour supprimer un canal de contact

L'`delete-contact-channel`exemple suivant supprime un canal de contact. La suppression d'un canal de contact garantit que le canal de contact ne sera pas paginé lors d'un incident.

```
aws ssm-contacts delete-contact-channel \
```

```
    --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/13149bad-52ee-45ea-ae1e-45857f78f9b2"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteContactChannelà la section Référence des AWS CLI commandes.

`delete-contact`

L'exemple de code suivant montre comment utiliser`delete-contact`.

AWS CLI

Pour supprimer un contact

L'`delete-contact`exemple suivant supprime un contact. Le contact ne sera plus joignable à partir d'un plan d'escalade qui le désigne.

```
aws ssm-contacts delete-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteContactà la section Référence des AWS CLI commandes.

`describe-engagement`

L'exemple de code suivant montre comment utiliser`describe-engagement`.

AWS CLI

Pour décrire les détails d'un engagement

L'`describe-engagement`exemple suivant répertorie les détails d'un engagement auprès d'un contact ou d'un plan d'escalade. Le sujet et le contenu sont envoyés aux canaux de contact.

```
aws ssm-contacts describe-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Sortie :

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
    "Sender": "cli",
    "Subject": "cli-test",
    "Content": "Testing engagements via CLI",
    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "StartTime": "2021-05-18T18:25:41.151000+00:00"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeEngagementà la section Référence des AWS CLI commandes.

`describe-page`

L'exemple de code suivant montre comment utiliser`describe-page`.

AWS CLI

Pour répertorier les détails d'une page sur un canal de contact

L'`describe-page`exemple suivant répertorie les détails d'une page sur un canal de contact. La page inclura le sujet et le contenu fournis.

```
aws ssm-contacts describe-page \
```

```
    --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93"
```

Sortie :

```
{
    "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93",
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0",
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "Sender": "cli",
    "Subject": "cli-test",
    "Content": "Testing engagements via CLI",
    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "SentTime": "2021-05-18T18:43:29.301000+00:00",
    "ReadTime": "2021-05-18T18:43:55.708000+00:00",
    "DeliveryTime": "2021-05-18T18:43:55.265000+00:00"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous DescribePageà la section Référence des AWS CLI commandes.

`get-contact-channel`

L'exemple de code suivant montre comment utiliser`get-contact-channel`.

AWS CLI

Pour répertorier les détails d'un canal de contact

L'`get-contact-channel`exemple suivant répertorie les détails d'un canal de contact.

```
aws ssm-contacts get-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

Sortie :

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "ContactChannelArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
    "Name": "akuas sms",
    "Type": "SMS",
    "DeliveryAddress": {
        "SimpleAddress": "+15005550199"
    },
    "ActivationStatus": "ACTIVATED"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetContactChannelà la section Référence des AWS CLI commandes.

`get-contact-policy`

L'exemple de code suivant montre comment utiliser`get-contact-policy`.

AWS CLI

Pour répertorier les politiques de ressources d'un contact

L'`get-contact-policy`exemple suivant répertorie les politiques de ressources associées au contact spécifié.

```
aws ssm-contacts get-contact-policy \
    --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Sortie :

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"SharePolicyForDocumentationDralia\",\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"222233334444\"},\"Action\":[\"ssm-contacts:GetContact\",\"ssm-
contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-
contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],\"Resource
```

```
\":[\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\",\"arn:aws:ssm-
contacts:*:111122223333:engagement/akuam/*\"]}]}"
}
```

Pour plus d'informations, consultez la section Utilisation de contacts partagés et de plans de réponse dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetContactPolicyà la section Référence des AWS CLI commandes.

`get-contact`

L'exemple de code suivant montre comment utiliser`get-contact`.

AWS CLI

Exemple 1 : Pour décrire un plan de contact

L'`get-contact`exemple suivant décrit un contact.

```
aws ssm-contacts get-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Sortie :

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "Alias": "akuam",
    "DisplayName": "Akua Mansa",
    "Type": "PERSONAL",
    "Plan": {
        "Stages": [
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ChannelTargetInfo": {
                            "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65",
                            "RetryIntervalInMinutes": 1
                        }
                    }
                ]
```

```
            },
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ChannelTargetInfo": {
                            "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad",
                            "RetryIntervalInMinutes": 1
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ChannelTargetInfo": {
                            "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/77d4f447-f619-4954-afff-85551e369c2a",
                            "RetryIntervalInMinutes": 1
                        }
                    }
                ]
            }
        ]
    }
}
```

Exemple 2 : Pour décrire un plan d'escalade

L'`get-contact`exemple suivant décrit un plan d'escalade.

```
aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"
```

Sortie :

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
    "Alias": "example_escalation",
```

```
    "DisplayName": "Example Escalation",
    "Type": "ESCALATION",
    "Plan": {
        "Stages": [
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ContactTargetInfo": {
                            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/akuam",
                            "IsEssential": true
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ContactTargetInfo": {
                            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/alejr",
                            "IsEssential": false
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 0,
                "Targets": [
                    {
                        "ContactTargetInfo": {
                            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/anasi",
                            "IsEssential": false
                        }
                    }
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous GetContactà la section Référence des AWS CLI commandes.

`list-contact-channels`

L'exemple de code suivant montre comment utiliser`list-contact-channels`.

AWS CLI

Pour répertorier les canaux de contact d'un contact

L'`list-contact-channels`exemple suivant répertorie les canaux de contact disponibles pour le contact spécifié.

```
aws ssm-contacts list-contact-channels \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Sortie :

```
{
    [
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Name": "akuas email",
            "Type": "EMAIL",
            "DeliveryAddress": {
                "SimpleAddress": "akuam@example.com"
            },
            "ActivationStatus": "NOT_ACTIVATED"
        },
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Name": "akuas sms",
            "Type": "SMS",
            "DeliveryAddress": {
```

```
                "SimpleAddress": "+15005550100"
            },
            "ActivationStatus": "ACTIVATED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListContactChannelsà la section Référence des AWS CLI commandes.

`list-contacts`

L'exemple de code suivant montre comment utiliser`list-contacts`.

AWS CLI

Pour répertorier tous les plans d'escalade et les contacts

L'`list-contacts`exemple suivant répertorie les contacts et les plans d'escalade de votre compte.

```
aws ssm-contacts list-contacts
```

Sortie :

```
{
    "Contacts": [
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Alias": "akuam",
            "DisplayName": "Akua Mansa",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
alejr",
            "Alias": "alejr",
```

```
            "DisplayName": "Alejandro Rosalez",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
anasi",
            "Alias": "anasi",
            "DisplayName": "Ana Carolina Silva",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
            "Alias": "example_escalation",
            "DisplayName": "Example Escalation",
            "Type": "ESCALATION"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListContactsà la section Référence des AWS CLI commandes.

`list-engagements`

L'exemple de code suivant montre comment utiliser`list-engagements`.

AWS CLI

Pour répertorier tous les engagements

L'`list-engagements`exemple suivant répertorie les engagements pris dans le cadre de plans d'escalade et de contacts. Vous pouvez également répertorier les engagements pour un seul incident.

```
aws ssm-contacts list-engagements
```

Sortie :

```
{
    "Engagements": [
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/91792571-0b53-4821-9f73-d25d13d9e529",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "StartTime": "2021-05-18T20:37:50.300000+00:00"
        },
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "StartTime": "2021-05-18T18:40:26.666000+00:00"
        },
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
            "Sender": "cli",
            "StartTime": "2021-05-18T18:25:41.151000+00:00"
        },
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "StartTime": "2021-05-18T18:20:58.093000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListEngagementsà la section Référence des AWS CLI commandes.

`list-page-receipts`

L'exemple de code suivant montre comment utiliser`list-page-receipts`.

AWS CLI

Pour répertorier les reçus de page

L'`command-name`exemple suivant indique si une page a été reçue ou non par un contact.

```
aws ssm-contacts list-page-receipts \
    --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"
```

Sortie :

```
{
    "Receipts": [
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ReceiptType": "DELIVERED",
            "ReceiptInfo": "425440",
            "ReceiptTime": "2021-05-18T20:42:57.485000+00:00"
        },
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ReceiptType": "READ",
            "ReceiptInfo": "425440",
            "ReceiptTime": "2021-05-18T20:42:57.907000+00:00"
        },
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ReceiptType": "SENT",
            "ReceiptInfo": "SM6656c19132f1465f9c9c1123a5dde7c9",
            "ReceiptTime": "2021-05-18T20:40:52.962000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListPageReceiptsà la section Référence des AWS CLI commandes.

`list-pages-by-contact`

L'exemple de code suivant montre comment utiliser`list-pages-by-contact`.

AWS CLI

Pour répertorier les pages par contact

L'`list-pages-by-contact`exemple suivant répertorie toutes les pages du contact spécifié.

```
aws ssm-contacts list-pages-by-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Sortie :

```
{
    "Pages": [
        {
            "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "SentTime": "2021-05-18T18:43:29.301000+00:00",
            "DeliveryTime": "2021-05-18T18:43:55.265000+00:00",
            "ReadTime": "2021-05-18T18:43:55.708000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListPagesByContactà la section Référence des AWS CLI commandes.

`list-pages-by-engagement`

L'exemple de code suivant montre comment utiliser`list-pages-by-engagement`.

AWS CLI

Répertorier les pages permettant de contacter les canaux créés à partir d'un engagement.

L'`list-pages-by-engagement`exemple suivant répertorie les pages qui se sont produites lors de l'engagement du plan d'engagement défini.

```
aws ssm-contacts list-pages-by-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0"
```

Sortie :

```
{
    "Pages": [
        {
            "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "SentTime": "2021-05-18T18:40:27.245000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListPagesByEngagementà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags d'un contact

L'`list-tags-for-resource`exemple suivant répertorie les tags du contact spécifié.

```
aws ssm-contacts list-tags-for-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "group1",
            "Value": "1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Balisage dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-contact-policy`

L'exemple de code suivant montre comment utiliser`put-contact-policy`.

AWS CLI

Pour partager un contact et des engagements

L'`put-contact-policy`exemple suivant ajoute une politique de ressources au contact Akua qui partage le contact et les engagements associés avec le principal.

```
aws ssm-contacts put-contact-policy \
    --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"ExampleResourcePolicy\",\"Action\":[\"ssm-contacts:GetContact\",\"ssm-
```

```
contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-
contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],
\"Principal\":{\"AWS\":\"222233334444\"},\"Effect\":\"Allow\",\"Resource
\":[\"arn:aws:ssm-contacts:*:111122223333:contact\/akuam\",\"arn:aws:ssm-
contacts:*:111122223333:engagement\/akuam\/*\"]}]}"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de contacts partagés et de plans de réponse dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous PutContactPolicyà la section Référence des AWS CLI commandes.

`send-activation-code`

L'exemple de code suivant montre comment utiliser`send-activation-code`.

AWS CLI

Pour envoyer un code d'activation

L'`send-activation-code`exemple suivant envoie un code d'activation et un message au canal de contact spécifié.

```
aws ssm-contacts send-activation-code \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/8ddae2d1-12c8-4e45-b852-c8587266c400"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous SendActivationCodeà la section Référence des AWS CLI commandes.

`start-engagement`

L'exemple de code suivant montre comment utiliser`start-engagement`.

AWS CLI

Exemple 1 : pour mettre en page les canaux de contact d'un contact

Les `start-engagement` pages suivantes permettent de contacter les canaux de contact. L'expéditeur, le sujet, le sujet public et le contenu public sont tous exempts de champs. Incident Manager envoie le sujet et le contenu aux canaux de contact VOICE ou EMAIL fournis. Incident Manager envoie le sujet public et le contenu public aux canaux de contact par SMS fournis. L'expéditeur est utilisé pour savoir qui a lancé l'engagement.

```
aws ssm-contacts start-engagement \
    --contact-id  "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \
    --sender "cli" \
    --subject "cli-test" \
    --content "Testing engagements via CLI" \
    --public-subject "cli-test" \
    --public-content "Testing engagements va CLI"
```

Sortie :

```
{
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

Exemple 2 : mettre en page un contact dans le plan d'escalade fourni.

Ce qui suit `start-engagement` engage les contacts par le biais d'un plan d'escalade. Chaque contact est paginé en fonction de son plan d'engagement.

```
aws ssm-contacts start-engagement \
    --contact-id  "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation" \
    --sender "cli" \
    --subject "cli-test" \
    --content "Testing engagements via CLI" \
    --public-subject "cli-test" \
```

```
    --public-content "Testing engagements va CLI"
```

Sortie :

```
{
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
}
```

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous StartEngagementà la section Référence des AWS CLI commandes.

`stop-engagement`

L'exemple de code suivant montre comment utiliser`stop-engagement`.

AWS CLI

Pour mettre fin à un engagement

L'`stop-engagement`exemple suivant empêche un engagement de paginer d'autres contacts et canaux de contact.

```
aws ssm-contacts stop-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous StopEngagementà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un contact

L'`tag-resource`exemple suivant étiquette un contact spécifié avec la paire clé-valeur de balise fournie.

```
aws ssm-contacts tag-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --tags '[{"Key":"group1","Value":"1"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les tags d'un contact

L'`untag-resource`exemple suivant supprime la balise group1 du contact spécifié.

```
aws ssm-contacts untag-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --tag-keys "group1"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-contact-channel`

L'exemple de code suivant montre comment utiliser`update-contact-channel`.

AWS CLI

Pour mettre à jour un canal de contact

L'`update-contact-channel`exemple suivant met à jour le nom et l'adresse de livraison d'un canal de contact.

```
aws ssm-contacts update-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad" \
    --name "akuas voice channel" \
    --delivery-address '{"SimpleAddress": "+15005550198"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateContactChannelà la section Référence des AWS CLI commandes.

`update-contact`

L'exemple de code suivant montre comment utiliser`update-contact`.

AWS CLI

Pour mettre à jour le plan d'engagement, contactez

L'`update-contact`exemple suivant met à jour le plan d'engagement du contact Akua pour inclure les trois types de canaux de contacts. Cela se fait après avoir créé des canaux de contact pour Akua.

```
aws ssm-contacts update-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \
    --plan '{"Stages": [{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo": {"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
```

```
channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65","RetryIntervalInMinutes":
 1 }}]}, {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":
{"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/
akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad", "RetryIntervalInMinutes": 1}}]},
 {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo": {"ContactChannelId":
 "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/77d4f447-
f619-4954-afff-85551e369c2a","RetryIntervalInMinutes": 1 }}]}]}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contacts dans le guide de l'utilisateur d'Incident Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateContactà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Inspector AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Inspector.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-attributes-to-findings`

L'exemple de code suivant montre comment utiliser`add-attributes-to-findings`.

AWS CLI

Pour ajouter des attributs aux résultats

La `add-attribute-to-finding` commande suivante affecte un attribut avec la clé `Example` et la valeur de `example` à la recherche avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU` :

```
aws inspector add-attributes-to-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attributes key=Example,value=example
```

Sortie :

```
{
    "failedItems": {}
}
```

Pour plus d'informations, consultez les conclusions d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous AddAttributesToFindingsà la section Référence des AWS CLI commandes.

`associate-member`

L'exemple de code suivant montre comment utiliser`associate-member`.

AWS CLI

Exemple : pour associer un AWS compte à un administrateur délégué Amazon Inspector

L'`associate-member`exemple suivant associe un AWS compte à un administrateur délégué Amazon Inspector.

```
aws inspector2 associate-member \
    --account-id 123456789012
```

Sortie :

```
{
    "accountId": "123456789012"
```

```
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes dans Amazon Inspector with AWS Organizations dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous AssociateMemberà la section Référence des AWS CLI commandes.

`create-assessment-target`

L'exemple de code suivant montre comment utiliser`create-assessment-target`.

AWS CLI

Pour créer un objectif d'évaluation

La `create-assessment-target` commande suivante crée une cible d'évaluation nommée `ExampleAssessmentTarget` à l'aide du groupe de ressources dont l'ARN est `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv` :

```
aws inspector create-assessment-target --assessment-target-
name ExampleAssessmentTarget --resource-group-arn arn:aws:inspector:us-
west-2:123456789012:resourcegroup/0-AB6DMKnv
```

Sortie :

```
{
    "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX"
}
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous CreateAssessmentTargetà la section Référence des AWS CLI commandes.

`create-assessment-template`

L'exemple de code suivant montre comment utiliser`create-assessment-template`.

AWS CLI

Pour créer un modèle d'évaluation

La `create-assessment-template` commande suivante crée un modèle d'évaluation appelé `ExampleAssessmentTemplate` pour la cible d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX` :

```
aws inspector create-assessment-template --assessment-target-
arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-
template-name ExampleAssessmentTemplate --duration-in-seconds 180 --rules-package-
arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p --user-
attributes-for-findings key=ExampleTag,value=examplevalue
```

Sortie :

```
{
    "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T"
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous CreateAssessmentTemplateà la section Référence des AWS CLI commandes.

`create-filter`

L'exemple de code suivant montre comment utiliser`create-filter`.

AWS CLI

Pour créer un filtre

L'`create-filter`exemple suivant crée une règle de suppression qui omet les résultats relatifs au type d'instance ECR.

```
aws inspector2 create-filter \
    --name "ExampleSuppressionRuleECR" \
    --description "This suppression rule omits ECR instance type findings" \
```

```
    --action SUPPRESS \
    --filter-criteria 'resourceType=[{comparison="EQUALS",
 value="AWS_ECR_INSTANCE"}]'
```

Sortie :

```
{
    "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
}
```

Pour plus d'informations, consultez Filtrer les résultats d'Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous CreateFilterà la section Référence des AWS CLI commandes.

`create-findings-report`

L'exemple de code suivant montre comment utiliser`create-findings-report`.

AWS CLI

Pour créer un rapport de conclusions

L'`create-findings-report`exemple suivant crée un rapport de recherche.

```
aws inspector2 create-findings-report \
    --report-format CSV \
    --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333 \
    --filter-criteria '{"ecrImageRepositoryName":
[{"comparison":"EQUALS","value":"debian"}]}'
```

Sortie :

```
{
    "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

Pour plus d'informations, consultez la section Gestion des résultats dans Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous CreateFindingsReportà la section Référence des AWS CLI commandes.

`create-resource-group`

L'exemple de code suivant montre comment utiliser`create-resource-group`.

AWS CLI

Pour créer un groupe de ressources

La `create-resource-group` commande suivante crée un groupe de ressources à l'aide de la clé de balise `Name` et de la valeur de `example` :

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

Sortie :

```
{
   "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-
AB6DMKnv"
}
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous CreateResourceGroupà la section Référence des AWS CLI commandes.

`create-sbom-export`

L'exemple de code suivant montre comment utiliser`create-sbom-export`.

AWS CLI

Pour créer un rapport de nomenclature logicielle (SBOM)

L'`create-sbom-export`exemple suivant crée un rapport de nomenclature logicielle (SBOM).

```
aws inspector2 create-sbom-export \
    --report-format SPDX_2_3 \
    --resource-filter-criteria
 'ecrRepositoryName=[{comparison="EQUALS",value="debian"}]' \
    --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333
```

Sortie :

```
{
    "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

Pour plus d'informations, consultez Exporter SBOMs avec Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous CreateSbomExportà la section Référence des AWS CLI commandes.

`delete-assessment-run`

L'exemple de code suivant montre comment utiliser`delete-assessment-run`.

AWS CLI

Pour supprimer un cycle d'évaluation

La `delete-assessment-run` commande suivante supprime l'évaluation exécutée avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe` :

```
aws inspector delete-assessment-run --assessment-run-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DeleteAssessmentRunà la section Référence des AWS CLI commandes.

`delete-assessment-target`

L'exemple de code suivant montre comment utiliser`delete-assessment-target`.

AWS CLI

Pour supprimer un objectif d'évaluation

La `delete-assessment-target` commande suivante supprime la cible d'évaluation dont l'ARN est : `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`

```
aws inspector delete-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DeleteAssessmentTargetà la section Référence des AWS CLI commandes.

`delete-assessment-template`

L'exemple de code suivant montre comment utiliser`delete-assessment-template`.

AWS CLI

Pour supprimer un modèle d'évaluation

La `delete-assessment-template` commande suivante supprime le modèle d'évaluation dont l'ARN est : `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T`

```
aws inspector delete-assessment-template --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DeleteAssessmentTemplateà la section Référence des AWS CLI commandes.

`delete-filter`

L'exemple de code suivant montre comment utiliser`delete-filter`.

AWS CLI

Pour supprimer un filtre

L'`delete-filter`exemple suivant supprime un filtre.

```
aws inspector2 delete-filter \
    --arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
```

Sortie :

```
{
    "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
}
```

Pour plus d'informations, consultez Filtrer les résultats d'Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DeleteFilterà la section Référence des AWS CLI commandes.

`describe-assessment-runs`

L'exemple de code suivant montre comment utiliser`describe-assessment-runs`.

AWS CLI

Pour décrire les cycles d'évaluation

La `describe-assessment-run` commande suivante décrit une évaluation exécutée avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE` :

```
aws inspector describe-assessment-runs --assessment-run-arns arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

Sortie :


```
{
        "assessmentRuns": [
          {
                "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE",
                "assessmentTemplateArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw",
                "completedAt": 1458680301.4,
                "createdAt": 1458680170.035,
                "dataCollected": true,
                "durationInSeconds": 3600,
                "name": "Run 1 for ExampleAssessmentTemplate",
                "notifications": [],
                "rulesPackageArns": [
                  "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
                ],
                "startedAt": 1458680170.161,
                "state": "COMPLETED",
                "stateChangedAt": 1458680301.4,
                "stateChanges": [
                  {
                        "state": "CREATED",
                        "stateChangedAt": 1458680170.035
                  },
                  {
                        "state": "START_DATA_COLLECTION_PENDING",
                        "stateChangedAt": 1458680170.065
                  },
                  {
                        "state": "START_DATA_COLLECTION_IN_PROGRESS",
                        "stateChangedAt": 1458680170.096
                  },
                  {
                        "state": "COLLECTING_DATA",
                        "stateChangedAt": 1458680170.161
                  },
                  {
                        "state": "STOP_DATA_COLLECTION_PENDING",
                        "stateChangedAt": 1458680239.883
                  },
                  {
                        "state": "DATA_COLLECTED",
```

```
                        "stateChangedAt": 1458680299.847
                    },
                    {
                        "state": "EVALUATING_RULES",
                        "stateChangedAt": 1458680300.099
                    },
                    {
                        "state": "COMPLETED",
                        "stateChangedAt": 1458680301.4
                    }
                ],
                "userAttributesForFindings": []
            }
        ],
        "failedItems": {}
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeAssessmentRunsà la section Référence des AWS CLI commandes.

`describe-assessment-targets`

L'exemple de code suivant montre comment utiliser`describe-assessment-targets`.

AWS CLI

Pour décrire les objectifs d'évaluation

La `describe-assessment-targets` commande suivante décrit la cible d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq` :

```
aws inspector describe-assessment-targets --assessment-target-
arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

Sortie :

```
{
      "assessmentTargets": [
        {
```

```
            "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq",
            "createdAt": 1458074191.459,
            "name": "ExampleAssessmentTarget",
            "resourceGroupArn": "arn:aws:inspector:us-
west-2:123456789012:resourcegroup/0-PyGXopAI",
            "updatedAt": 1458074191.459
        }
      ],
      "failedItems": {}
}
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeAssessmentTargetsà la section Référence des AWS CLI commandes.

`describe-assessment-templates`

L'exemple de code suivant montre comment utiliser`describe-assessment-templates`.

AWS CLI

Pour décrire les modèles d'évaluation

La `describe-assessment-templates` commande suivante décrit le modèle d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw` :

```
aws inspector describe-assessment-templates --assessment-template-
arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw
```

Sortie :

```
{
      "assessmentTemplates": [
        {
            "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw",
            "assessmentTargetArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq",
```

```
            "createdAt": 1458074191.844,
            "durationInSeconds": 3600,
            "name": "ExampleAssessmentTemplate",
            "rulesPackageArns": [
              "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
            ],
            "userAttributesForFindings": []
        }
      ],
      "failedItems": {}
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeAssessmentTemplatesà la section Référence des AWS CLI commandes.

`describe-cross-account-access-role`

L'exemple de code suivant montre comment utiliser`describe-cross-account-access-role`.

AWS CLI

Pour décrire le rôle d'accès entre comptes

La `describe-cross-account-access-role` commande suivante décrit le rôle IAM qui permet à Amazon Inspector d'accéder à votre AWS compte :

```
aws inspector describe-cross-account-access-role
```

Sortie :

```
{
        "registeredAt": 1458069182.826,
        "roleArn": "arn:aws:iam::123456789012:role/inspector",
        "valid": true
}
```

Pour plus d'informations, consultez la section Configuration d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeCrossAccountAccessRoleà la section Référence des AWS CLI commandes.

`describe-findings`

L'exemple de code suivant montre comment utiliser`describe-findings`.

AWS CLI

Pour décrire les résultats

La `describe-findings` commande suivante décrit le résultat obtenu avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4` :

```
aws inspector describe-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4
```

Sortie :

```
{
      "failedItems": {},
      "findings": [
        {
              "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
              "assetAttributes": {
                "ipv4Addresses": [],
                "schemaVersion": 1
              },
              "assetType": "ec2-instance",
              "attributes": [],
              "confidence": 10,
              "createdAt": 1458680301.37,
              "description": "Amazon Inspector did not find any potential security
 issues during this assessment.",
              "indicatorOfCompromise": false,
              "numericSeverity": 0,
              "recommendation": "No remediation needed.",
              "schemaVersion": 1,
              "service": "Inspector",
```

```
            "serviceAttributes": {
              "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
              "rulesPackageArn": "arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-X1KXtawP",
              "schemaVersion": 1
            },
            "severity": "Informational",
            "title": "No potential security issues found",
            "updatedAt": 1458680301.37,
            "userAttributes": []
        }
    ]
}
```

Pour plus d'informations, consultez les conclusions d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeFindingsà la section Référence des AWS CLI commandes.

`describe-resource-groups`

L'exemple de code suivant montre comment utiliser`describe-resource-groups`.

AWS CLI

Pour décrire les groupes de ressources

La `describe-resource-groups` commande suivante décrit le groupe de ressources dont l'ARN est `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI` :

```
aws inspector describe-resource-groups --resource-group-arns arn:aws:inspector:us-
west-2:123456789012:resourcegroup/0-PyGXopAI
```

Sortie :

```
{
    "failedItems": {},
    "resourceGroups": [
```

```
        {
              "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-
PyGXopAI",
              "createdAt": 1458074191.098,
              "tags": [
                {
                      "key": "Name",
                      "value": "example"
                }
              ]
        }
      ]
}
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeResourceGroupsà la section Référence des AWS CLI commandes.

`describe-rules-packages`

L'exemple de code suivant montre comment utiliser`describe-rules-packages`.

AWS CLI

Pour décrire les packages de règles

La `describe-rules-packages` commande suivante décrit le package de règles avec l'ARN de `arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p` :

```
aws inspector describe-rules-packages --rules-package-arns arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-9hgA516p
```

Sortie :

```
{
      "failedItems": {},
      "rulesPackages": [
        {
              "arn": "arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-9hgA516p",
```

```
            "description": "The rules in this package help verify whether the EC2
 instances in your application are exposed to Common Vulnerabilities and
            Exposures (CVEs). Attacks can exploit unpatched vulnerabilities to
 compromise the confidentiality, integrity, or availability of your service
            or data. The CVE system provides a reference for publicly known
 information security vulnerabilities and exposures. For more information, see
            [https://cve.mitre.org/](https://cve.mitre.org/). If a particular CVE
 appears in one of the produced Findings at the end of a completed
            Inspector assessment, you can search [https://cve.mitre.org/](https://
cve.mitre.org/) using the CVE's ID (for example, \"CVE-2009-0021\") to
            find detailed information about this CVE, its severity, and how to
 mitigate it. ",
            "name": "Common Vulnerabilities and Exposures",
            "provider": "Amazon Web Services, Inc.",
            "version": "1.1"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Inspector Rules Packages and Rules dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DescribeRulesPackagesà la section Référence des AWS CLI commandes.

`disassociate-member`

L'exemple de code suivant montre comment utiliser`disassociate-member`.

AWS CLI

Exemple : pour dissocier un compte membre d'un administrateur délégué Amazon Inspector

L'`disassociate-member`exemple suivant dissocie un AWS compte d'un administrateur délégué Amazon Inspector.

```
aws inspector2 disassociate-member \
    --account-id 123456789012
```

Sortie :

```
{
    "accountId": "123456789012"
```

```
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes dans Amazon Inspector with AWS Organizations dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous DisassociateMemberà la section Référence des AWS CLI commandes.

`get-configuration`

L'exemple de code suivant montre comment utiliser`get-configuration`.

AWS CLI

Pour obtenir la configuration des paramètres pour les scans Inspector

L'`get-configuration`exemple suivant permet d'obtenir la configuration des paramètres pour les scans Inspector.

```
aws inspector2 get-configuration
```

Sortie :

```
{
    "ec2Configuration": {
        "scanModeState": {
            "scanMode": "EC2_HYBRID",
            "scanModeStatus": "SUCCESS"
        }
    },
    "ecrConfiguration": {
        "rescanDurationState": {
            "pullDateRescanDuration": "DAYS_90",
            "rescanDuration": "DAYS_30",
            "status": "SUCCESS",
            "updatedAt": "2024-05-14T21:16:20.237000+00:00"
        }
     }
}
```

Pour plus d'informations, consultez la section Analyse automatisée des ressources avec Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous GetConfigurationà la section Référence des AWS CLI commandes.

`get-member`

L'exemple de code suivant montre comment utiliser`get-member`.

AWS CLI

Exemple : pour obtenir des informations sur les membres de votre organisation

aws inspector2 get-member --account-id 123456789012

Sortie :

```
{
        "member": {
        "accountId": "123456789012",
        "delegatedAdminAccountId": "123456789012",
        "relationshipStatus": "ENABLED",
        "updatedAt": "2023-09-11T09:57:20.520000-07:00"
    }
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes dans Amazon Inspector with AWS Organizations dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous GetMemberà la section Référence des AWS CLI commandes.

`get-telemetry-metadata`

L'exemple de code suivant montre comment utiliser`get-telemetry-metadata`.

AWS CLI

Pour obtenir les métadonnées de télémétrie

La `get-telemetry-metadata` commande suivante génère des informations sur les données collectées pour l'évaluation exécutée avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE` :

```
aws inspector get-telemetry-metadata --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

Sortie :

```
{
      "telemetryMetadata": [
        {
              "count": 2,
              "dataSize": 345,
              "messageType": "InspectorDuplicateProcess"
        },
        {
              "count": 3,
              "dataSize": 255,
              "messageType": "InspectorTimeEventMsg"
        },
        {
              "count": 4,
              "dataSize": 1082,
              "messageType": "InspectorNetworkInterface"
        },
        {
              "count": 2,
              "dataSize": 349,
              "messageType": "InspectorDnsEntry"
        },
        {
              "count": 11,
              "dataSize": 2514,
              "messageType": "InspectorDirectoryInfoMsg"
        },
        {
              "count": 1,
              "dataSize": 179,
              "messageType": "InspectorTcpV6ListeningPort"
        },
        {
              "count": 101,
              "dataSize": 10949,
              "messageType": "InspectorTerminal"
        },
        {
```

```
            "count": 26,
            "dataSize": 5916,
            "messageType": "InspectorUser"
        },
        {
            "count": 282,
            "dataSize": 32148,
            "messageType": "InspectorDynamicallyLoadedCodeModule"
        },
        {
            "count": 18,
            "dataSize": 10172,
            "messageType": "InspectorCreateProcess"
        },
        {
            "count": 3,
            "dataSize": 8001,
            "messageType": "InspectorProcessPerformance"
        },
        {
            "count": 1,
            "dataSize": 360,
            "messageType": "InspectorOperatingSystem"
        },
        {
            "count": 6,
            "dataSize": 546,
            "messageType": "InspectorStopProcess"
        },
        {
            "count": 1,
            "dataSize": 1553,
            "messageType": "InspectorInstanceMetaData"
        },
        {
            "count": 2,
            "dataSize": 434,
            "messageType": "InspectorTcpV4Connection"
        },
        {
            "count": 474,
            "dataSize": 2960322,
            "messageType": "InspectorPackageInfo"
        },
```

```
        {
              "count": 3,
              "dataSize": 2235,
              "messageType": "InspectorSystemPerformance"
        },
        {
              "count": 105,
              "dataSize": 46048,
              "messageType": "InspectorCodeModule"
        },
        {
              "count": 1,
              "dataSize": 182,
              "messageType": "InspectorUdpV6ListeningPort"
        },
        {
              "count": 2,
              "dataSize": 371,
              "messageType": "InspectorUdpV4ListeningPort"
        },
        {
              "count": 18,
              "dataSize": 8362,
              "messageType": "InspectorKernelModule"
        },
        {
              "count": 29,
              "dataSize": 48788,
              "messageType": "InspectorConfigurationInfo"
        },
        {
              "count": 1,
              "dataSize": 79,
              "messageType": "InspectorMonitoringStart"
        },
        {
              "count": 5,
              "dataSize": 0,
              "messageType": "InspectorSplitMsgBegin"
        },
        {
              "count": 51,
              "dataSize": 4593,
              "messageType": "InspectorGroup"
```

```
        },
        {
              "count": 1,
              "dataSize": 184,
              "messageType": "InspectorTcpV4ListeningPort"
        },
        {
              "count": 1159,
              "dataSize": 3146579,
              "messageType": "Total"
        },
        {
              "count": 5,
              "dataSize": 0,
              "messageType": "InspectorSplitMsgEnd"
        },
        {
              "count": 1,
              "dataSize": 612,
              "messageType": "InspectorLoadImageInProcess"
        }
      ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetTelemetryMetadataà la section Référence des AWS CLI commandes.

`list-account-permissions`

L'exemple de code suivant montre comment utiliser`list-account-permissions`.

AWS CLI

Pour répertorier les autorisations du compte

L'`list-account-permissions`exemple suivant répertorie les autorisations de votre compte.

```
aws inspector2 list-account-permissions
```

Sortie :

```
{
```

```
    "permissions": [
        {
            "operation": "ENABLE_SCANNING",
            "service": "ECR"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "ECR"
        },
        {
            "operation": "ENABLE_REPOSITORY",
            "service": "ECR"
        },
        {
            "operation": "DISABLE_REPOSITORY",
            "service": "ECR"
        },
        {
            "operation": "ENABLE_SCANNING",
            "service": "EC2"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "EC2"
        },
        {
            "operation": "ENABLE_SCANNING",
            "service": "LAMBDA"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "LAMBDA"
        }
    ]
}
```

Pour plus d'informations, consultez Identity and Access Management for Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListAccountPermissionsà la section Référence des AWS CLI commandes.

`list-assessment-run-agents`

L'exemple de code suivant montre comment utiliser`list-assessment-run-agents`.

AWS CLI

Pour répertorier les agents d'évaluation, exécutez

La `list-assessment-run-agents` commande suivante répertorie les agents de l'évaluation exécutée avec l'ARN spécifié.

```
aws inspector list-assessment-run-agents \
    --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE
```

Sortie :

```
{
    "assessmentRunAgents": [
        {
            "agentHealth": "HEALTHY",
            "agentHealthCode": "HEALTHY",
            "agentId": "i-49113b93",
            "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
            "telemetryMetadata": [
                {
                    "count": 2,
                    "dataSize": 345,
                    "messageType": "InspectorDuplicateProcess"
                },
                {
                    "count": 3,
                    "dataSize": 255,
                    "messageType": "InspectorTimeEventMsg"
                },
                {
                    "count": 4,
                    "dataSize": 1082,
                    "messageType": "InspectorNetworkInterface"
                },
                {
                    "count": 2,
```

```
                    "dataSize": 349,
                    "messageType": "InspectorDnsEntry"
                },
                {
                    "count": 11,
                    "dataSize": 2514,
                    "messageType": "InspectorDirectoryInfoMsg"
                },
                {
                    "count": 1,
                    "dataSize": 179,
                    "messageType": "InspectorTcpV6ListeningPort"
                },
                {
                    "count": 101,
                    "dataSize": 10949,
                    "messageType": "InspectorTerminal"
                },
                {
                    "count": 26,
                    "dataSize": 5916,
                    "messageType": "InspectorUser"
                },
                {
                    "count": 282,
                    "dataSize": 32148,
                    "messageType": "InspectorDynamicallyLoadedCodeModule"
                },
                {
                    "count": 18,
                    "dataSize": 10172,
                    "messageType": "InspectorCreateProcess"
                },
                {
                    "count": 3,
                    "dataSize": 8001,
                    "messageType": "InspectorProcessPerformance"
                },
                {
                    "count": 1,
                    "dataSize": 360,
                    "messageType": "InspectorOperatingSystem"
                },
                {
```

```
                    "count": 6,
                    "dataSize": 546,
                    "messageType": "InspectorStopProcess"
                },
                {
                    "count": 1,
                    "dataSize": 1553,
                    "messageType": "InspectorInstanceMetaData"
                },
                {
                    "count": 2,
                    "dataSize": 434,
                    "messageType": "InspectorTcpV4Connection"
                },
                {
                    "count": 474,
                    "dataSize": 2960322,
                    "messageType": "InspectorPackageInfo"
                },
                {
                    "count": 3,
                    "dataSize": 2235,
                    "messageType": "InspectorSystemPerformance"
                },
                {
                    "count": 105,
                    "dataSize": 46048,
                    "messageType": "InspectorCodeModule"
                },
                {
                    "count": 1,
                    "dataSize": 182,
                    "messageType": "InspectorUdpV6ListeningPort"
                },
                {
                    "count": 2,
                    "dataSize": 371,
                    "messageType": "InspectorUdpV4ListeningPort"
                },
                {
                    "count": 18,
                    "dataSize": 8362,
                    "messageType": "InspectorKernelModule"
                },
```

```
                {
                    "count": 29,
                    "dataSize": 48788,
                    "messageType": "InspectorConfigurationInfo"
                },
                {
                    "count": 1,
                    "dataSize": 79,
                    "messageType": "InspectorMonitoringStart"
                },
                {
                    "count": 5,
                    "dataSize": 0,
                    "messageType": "InspectorSplitMsgBegin"
                },
                {
                    "count": 51,
                    "dataSize": 4593,
                    "messageType": "InspectorGroup"
                },
                {
                    "count": 1,
                    "dataSize": 184,
                    "messageType": "InspectorTcpV4ListeningPort"
                },
                {
                    "count": 1159,
                    "dataSize": 3146579,
                    "messageType": "Total"
                },
                {
                    "count": 5,
                    "dataSize": 0,
                    "messageType": "InspectorSplitMsgEnd"
                },
                {
                    "count": 1,
                    "dataSize": 612,
                    "messageType": "InspectorLoadImageInProcess"
                }
            ]
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section AWS Agents dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListAssessmentRunAgentsà la section Référence des AWS CLI commandes.

`list-assessment-runs`

L'exemple de code suivant montre comment utiliser`list-assessment-runs`.

AWS CLI

Pour répertorier les cycles d'évaluation

La `list-assessment-runs` commande suivante répertorie toutes les séries d'évaluation existantes.

```
aws inspector list-assessment-runs
```

Sortie :

```
{
    "assessmentRunArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE",
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v"
    ]
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListAssessmentRunsà la section Référence des AWS CLI commandes.

`list-assessment-targets`

L'exemple de code suivant montre comment utiliser`list-assessment-targets`.

AWS CLI

Pour répertorier les cibles d'évaluation

La `list-assessment-targets` commande suivante répertorie toutes les cibles d'évaluation existantes :

```
aws inspector list-assessment-targets
```

Sortie :

```
{
       "assessmentTargetArns": [
       "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
       ]
}
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListAssessmentTargetsà la section Référence des AWS CLI commandes.

`list-assessment-templates`

L'exemple de code suivant montre comment utiliser`list-assessment-templates`.

AWS CLI

Pour répertorier les modèles d'évaluation

La `list-assessment-templates` commande suivante répertorie tous les modèles d'évaluation existants :

```
aws inspector list-assessment-templates
```

Sortie :

```
{
       "assessmentTemplateArns": [
       "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw",
```

```
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-
Uza6ihLh"
        ]
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListAssessmentTemplatesà la section Référence des AWS CLI commandes.

`list-coverage-statistics`

L'exemple de code suivant montre comment utiliser`list-coverage-statistics`.

AWS CLI

Exemple 1 : pour répertorier les statistiques de couverture par groupes

L'`list-coverage-statistics`exemple suivant répertorie les statistiques de couverture de votre AWS environnement par groupes.

```
aws inspector2 list-coverage-statistics \
   --group-by RESOURCE_TYPE
```

Sortie :

```
{
    "countsByGroup": [
        {
            "count": 56,
            "groupKey": "AWS_LAMBDA_FUNCTION"
        },
        {
            "count": 27,
            "groupKey": "AWS_ECR_REPOSITORY"
        },
        {
            "count": 18,
            "groupKey": "AWS_EC2_INSTANCE"
        },
        {
            "count": 3,
```

```
            "groupKey": "AWS_ECR_CONTAINER_IMAGE"
        },
        {
            "count": 1,
            "groupKey": "AWS_ACCOUNT"
        }
    ],
    "totalCounts": 105
}
```

Pour plus d'informations, consultez la section Évaluation de la couverture de votre AWS environnement par Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

Exemple 2 : pour répertorier les statistiques de couverture par type de ressource

L'`list-coverage-statistics`exemple suivant répertorie les statistiques de couverture de votre AWS environnement par type de ressource.

```
aws inspector2 list-coverage-statistics
    --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_ECR_REPOSITORY"}]}'
    --group-by SCAN_STATUS_REASON
```

Sortie :

```
{
    "countsByGroup": [
        {
            "count": 27,
            "groupKey": "SUCCESSFUL"
        }
    ],
    "totalCounts": 27
}
```

Pour plus d'informations, consultez la section Évaluation de la couverture de votre AWS environnement par Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

Exemple 3 : pour répertorier les statistiques de couverture par nom de référentiel ECR

L'`list-coverage-statistics`exemple suivant répertorie les statistiques de couverture de votre AWS environnement par nom de référentiel ECR.

```
aws inspector2 list-coverage-statistics
   --filter-criteria '{"ecrRepositoryName":
[{"comparison":"EQUALS","value":"debian"}]}'
   --group-by SCAN_STATUS_REASON
```

Sortie :

```
{
    "countsByGroup": [
        {
            "count": 3,
            "groupKey": "SUCCESSFUL"
        }
    ],
    "totalCounts": 3
}
```

Pour plus d'informations, consultez la section Évaluation de la couverture de votre AWS environnement par Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListCoverageStatisticsà la section Référence des AWS CLI commandes.

`list-coverage`

L'exemple de code suivant montre comment utiliser`list-coverage`.

AWS CLI

Exemple 1 : pour répertorier les détails de couverture relatifs à votre environnement

L'`list-coverage`exemple suivant répertorie les détails de couverture de votre environnement.

```
aws inspector2 list-coverage
```

Sortie :

```
{
    "coveredResources": [
        {
            "accountId": "123456789012",
```

```
            "lastScannedAt": "2024-05-20T16:23:20-07:00",
            "resourceId": "i-EXAMPLE55555555555",
            "resourceMetadata": {
                "ec2": {
                    "amiId": "ami-EXAMPLE6666666666",
                    "platform": "LINUX"
                }
            },
            "resourceType": "AWS_EC2_INSTANCE",
            "scanStatus": {
                "reason": "SUCCESSFUL",
                "statusCode": "ACTIVE"
            },
            "scanType": "PACKAGE"
        }
    ]
}
```

Exemple 2 : pour répertorier les détails de couverture relatifs au type de ressource de la fonction Lambda

L'`list-coverage`exemple suivant répertorie les détails du type de ressource de votre fonction Lamda.

```
aws inspector2 list-coverage
    --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_LAMBDA_FUNCTION"}]}'
```

Sortie :

```
{
    "coveredResources": [
        {
            "accountId": "123456789012",
            "resourceId": "arn:aws:lambda:us-west-2:123456789012:function:Eval-
container-scan-results:$LATEST",
            "resourceMetadata": {
                "lambdaFunction": {
                    "functionName": "Eval-container-scan-results",
                    "functionTags": {},
                    "layers": [],
                    "runtime": "PYTHON_3_7"
                }
```

```
            },
            "resourceType": "AWS_LAMBDA_FUNCTION",
            "scanStatus": {
                "reason": "SUCCESSFUL",
                "statusCode": "ACTIVE"
            },
            "scanType": "CODE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListCoverageà la section Référence des AWS CLI commandes.

`list-delegated-admin-accounts`

L'exemple de code suivant montre comment utiliser`list-delegated-admin-accounts`.

AWS CLI

Pour répertorier les informations relatives au compte d'administrateur délégué de votre organisation

L'`list-delegated-admin-accounts`exemple suivant répertorie les informations relatives au compte d'administrateur délégué de votre organisation.

```
aws inspector2 list-delegated-admin-accounts
```

Sortie :

```
{
    "delegatedAdminAccounts": [
        {
            "accountId": "123456789012",
            "status": "ENABLED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Désignation d'un administrateur délégué pour Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListDelegatedAdminAccountsà la section Référence des AWS CLI commandes.

`list-event-subscriptions`

L'exemple de code suivant montre comment utiliser`list-event-subscriptions`.

AWS CLI

Pour répertorier les abonnements aux événements

La `list-event-subscriptions` commande suivante répertorie tous les abonnements aux événements pour le modèle d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0` :

```
aws inspector list-event-subscriptions --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0
```

Sortie :

```
{
      "subscriptions": [
        {
              "eventSubscriptions": [
                {
                      "event": "ASSESSMENT_RUN_COMPLETED",
                      "subscribedAt": 1459455440.867
                }
              ],
              "resourceArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0",
              "topicArn": "arn:aws:sns:us-west-2:123456789012:exampletopic"
        }
      ]
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListEventSubscriptionsà la section Référence des AWS CLI commandes.

`list-filters`

L'exemple de code suivant montre comment utiliser`list-filters`.

AWS CLI

Pour répertorier les filtres associés au compte que vous avez utilisé pour activer Amazon Inspector

Les `list-filters` exemples suivants répertorient les filtres associés au compte que vous avez utilisé pour activer Amazon Inspector.

```
aws inspector2 list-filters
```

Sortie :

```
{
    "filters": [
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:11:08.602000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_EC2_INSTANCE"
                    },
                ]
            },
            "description": "This suppression rule omits EC2 instance type findings",
            "name": "ExampleSuppressionRuleEC2",
            "ownerId": "o-EXAMPLE222",
            "tags": {},
            "updatedAt": "2024-05-15T21:11:08.602000+00:00"
        },
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-east-1:813737243517:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:28:27.054000+00:00",
            "criteria": {
```

```
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_ECR_INSTANCE"
                    }
                ]
            },
            "description": "This suppression rule omits ECR instance type findings",
            "name": "ExampleSuppressionRuleECR",
            "ownerId": "o-EXAMPLE222",
            "tags": {},
            "updatedAt": "2024-05-15T21:28:27.054000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez Filtrer les résultats d'Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListFiltersà la section Référence des AWS CLI commandes.

`list-findings`

L'exemple de code suivant montre comment utiliser`list-findings`.

AWS CLI

Pour répertorier les résultats

La `list-findings` commande suivante répertorie tous les résultats générés :

```
aws inspector list-findings
```

Sortie :

```
{
        "findingArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v/finding/0-tyvmqBLy"
```

```
        ]
}
```

Pour plus d'informations, consultez les conclusions d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListFindingsà la section Référence des AWS CLI commandes.

`list-members`

L'exemple de code suivant montre comment utiliser`list-members`.

AWS CLI

Exemple 1 : pour répertorier tous les comptes membres associés à l'administrateur délégué Amazon Inspector de votre organisation

aws inspector2 list-members --only-associated

Sortie :

```
{
        {
             "members": [
        {
             "accountId": "123456789012",
             "delegatedAdminAccountId": "123456789012",
             "relationshipStatus": "ENABLED",
             "updatedAt": "2023-09-11T09:57:20.520000-07:00"
        },
        {
             "accountId": "123456789012",
             "delegatedAdminAccountId": "123456789012",
             "relationshipStatus": "ENABLED",
             "updatedAt": "2024-08-12T10:13:01.472000-07:00"
        },
        {
             "accountId": "625032911453",
             "delegatedAdminAccountId": "123456789012",
             "relationshipStatus": "ENABLED",
             "updatedAt": "2023-09-11T09:57:20.438000-07:00"
        },
```

```
        {
            "accountId": "715411239211",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-04-24T09:14:57.471000-07:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes dans Amazon Inspector with AWS Organizations dans le guide de l'utilisateur d'Amazon Inspector.

Exemple 2 : pour répertorier tous les comptes membres associés et dissociés de l'administrateur délégué Amazon Inspector de votre organisation

membres de la liste des membres d'AWS Inspector2 -- no-only-associated

Sortie :

```
{
        {
            "members": [
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "REMOVED",
            "updatedAt": "2024-05-15T11:34:53.326000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.520000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-08-12T10:13:01.472000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
```

```
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.438000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-04-24T09:14:57.471000-07:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes dans Amazon Inspector with AWS Organizations dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListMembersà la section Référence des AWS CLI commandes.

`list-rules-packages`

L'exemple de code suivant montre comment utiliser`list-rules-packages`.

AWS CLI

Pour répertorier les packages de règles

La `list-rules-packages` commande suivante répertorie tous les packages de règles Inspector disponibles :

```
aws inspector list-rules-packages
```

Sortie :

```
{
       "rulesPackageArns": [
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-H5hpSawc",
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-JJOtZiqQ",
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-vg5GGHSD"
       ]
 }
```

Pour plus d'informations, consultez Amazon Inspector Rules Packages and Rules dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListRulesPackagesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

La `list-tags-for-resource` commande suivante répertorie toutes les balises associées au modèle d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu` :

```
aws inspector list-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu
```

Sortie :

```
{
      "tags": [
        {
              "key": "Name",
              "value": "Example"
        }
      ]
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-usage-totals`

L'exemple de code suivant montre comment utiliser`list-usage-totals`.

AWS CLI

Pour répertorier les totaux d'utilisation au cours des 30 derniers jours

Les `list-usage-totals` exemples suivants répertorient les totaux d'utilisation au cours des 30 derniers jours.

```
aws inspector2 list-usage-totals
```

Sortie :

```
{
    "totals": [
        {
            "accountId": "123456789012",
            "usage": [
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 4.6022044647,
                    "total": 1893.4784083333334,
                    "type": "EC2_AGENTLESS_INSTANCE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 18.892449279,
                    "total": 10882.050784722222,
                    "type": "EC2_INSTANCE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 5.4525363736,
                    "total": 6543.043648333333,
                    "type": "LAMBDA_FUNCTION_CODE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 3.9064080309,
                    "total": 9375.379274166668,
                    "type": "LAMBDA_FUNCTION_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 0.06,
```

```
                    "total": 6.0,
                    "type": "ECR_RESCAN"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 0.09,
                    "total": 1.0,
                    "type": "ECR_INITIAL_SCAN"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance de l'utilisation et des coûts dans Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous ListUsageTotalsà la section Référence des AWS CLI commandes.

`preview-agents`

L'exemple de code suivant montre comment utiliser`preview-agents`.

AWS CLI

Pour prévisualiser les agents

La `preview-agents` commande suivante affiche un aperçu des agents installés sur les EC2 instances faisant partie de la cible d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq` :

```
aws inspector preview-agents --preview-agents-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

Sortie :

```
{
      "agentPreviews": [
        {
              "agentId": "i-49113b93"
        }
```

```
        ]
}
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous PreviewAgentsà la section Référence des AWS CLI commandes.

`register-cross-account-access-role`

L'exemple de code suivant montre comment utiliser`register-cross-account-access-role`.

AWS CLI

Pour enregistrer le rôle d'accès multicompte

La `register-cross-account-access-role` commande suivante enregistre le rôle IAM auprès de `arn:aws:iam::123456789012:role/inspector` l'ARN utilisé par Amazon Inspector pour répertorier vos EC2 instances au début de l'exécution de l'évaluation ou lorsque vous appelez la commande preview-agents :

```
aws inspector register-cross-account-access-role --role-
arn arn:aws:iam::123456789012:role/inspector
```

Pour plus d'informations, consultez la section Configuration d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous RegisterCrossAccountAccessRoleà la section Référence des AWS CLI commandes.

`remove-attributes-from-findings`

L'exemple de code suivant montre comment utiliser`remove-attributes-from-findings`.

AWS CLI

Pour supprimer des attributs des résultats

La `remove-attributes-from-finding` commande suivante supprime l'attribut avec la clé `Example` et la valeur de de `example` de la recherche avec l'ARN de `arn:aws:inspector:us-`

`west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU` :

```
aws inspector remove-attributes-from-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attribute-keys key=Example,value=example
```

Sortie :

```
{
      "failedItems": {}
}
```

Pour plus d'informations, consultez les conclusions d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous RemoveAttributesFromFindingsà la section Référence des AWS CLI commandes.

`set-tags-for-resource`

L'exemple de code suivant montre comment utiliser`set-tags-for-resource`.

AWS CLI

Pour définir des balises pour une ressource

La `set-tags-for-resource` commande suivante définit la balise avec la clé `Example` et la valeur `example` de sur le modèle d'évaluation dont l'ARN est `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0` :

```
aws inspector set-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0 --tags key=Example,value=example
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous SetTagsForResourceà la section Référence des AWS CLI commandes.

`start-assessment-run`

L'exemple de code suivant montre comment utiliser`start-assessment-run`.

AWS CLI

Pour démarrer un cycle d'évaluation

La `start-assessment-run` commande suivante lance l'exécution d'évaluation nommée `examplerun` à l'aide du modèle d'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T` :

```
aws inspector start-assessment-run --assessment-run-name examplerun --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

Sortie :

```
{
      "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY"
}
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous StartAssessmentRunà la section Référence des AWS CLI commandes.

`stop-assessment-run`

L'exemple de code suivant montre comment utiliser`stop-assessment-run`.

AWS CLI

Pour arrêter un cycle d'évaluation

La `stop-assessment-run` commande suivante arrête l'exécution de l'évaluation avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY` :

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous StopAssessmentRunà la section Référence des AWS CLI commandes.

`subscribe-to-event`

L'exemple de code suivant montre comment utilisersubscribe-to-event.

AWS CLI

Pour s'abonner à un événement

L'exemple suivant permet d'envoyer des notifications Amazon SNS concernant l'ASSESSMENT_RUN_COMPLETEDévénement au sujet avec l'ARN de arn:aws:sns:us-west-2:123456789012:exampletopic

```
aws inspector subscribe-to-event \
    --event ASSESSMENT_RUN_COMPLETED \
    --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/
template/0-7sbz2Kz0 \
    --topic-arn arn:aws:sns:us-west-2:123456789012:exampletopic
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous SubscribeToEventà la section Référence des AWS CLI commandes.

`unsubscribe-from-event`

L'exemple de code suivant montre comment utiliserunsubscribe-from-event.

AWS CLI

Pour se désinscrire d'un événement

La `unsubscribe-from-event` commande suivante désactive le processus d'envoi de notifications Amazon SNS concernant `ASSESSMENT_RUN_COMPLETED` l'événement au sujet avec l'ARN suivant : `arn:aws:sns:us-west-2:123456789012:exampletopic`

```
aws inspector unsubscribe-from-event --event ASSESSMENT_RUN_COMPLETED --resource-
arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0
 --topic arn:aws:sns:us-west-2:123456789012:exampletopic
```

Pour plus d'informations, consultez les modèles d'évaluation et les cycles d'évaluation Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous UnsubscribeFromEventà la section Référence des AWS CLI commandes.

`update-assessment-target`

L'exemple de code suivant montre comment utiliser`update-assessment-target`.

AWS CLI

Pour mettre à jour un objectif d'évaluation

La `update-assessment-target` commande suivante met à jour la cible d'évaluation avec l'ARN `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX` et le nom de`Example`, et le groupe de ressources avec l'ARN de `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt` :

```
aws inspector update-assessment-target --assessment-target-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX --assessment-target-name Example --resource-
group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt
```

Pour plus d'informations, consultez les cibles d'évaluation d'Amazon Inspector dans le guide Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous UpdateAssessmentTargetà la section Référence des AWS CLI commandes.

`update-filter`

L'exemple de code suivant montre comment utiliser`update-filter`.

AWS CLI

Pour mettre à jour un filtre

L'`update-filter`exemple suivant met à jour un filtre pour omettre les résultats Lambda au lieu des résultats de l'instance ECR.

```
aws inspector2 update-filter \
   --filter-arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444" \
   --name "ExampleSuppressionRuleLambda" \
   --description "This suppression rule omits Lambda instance findings" \
   --reason "Updating filter to omit Lambda instance findings instead of ECR
 instance findings"
```

Sortie :

```
{
    "filters": [
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:28:27.054000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_ECR_INSTANCE"
                    }
                ]
            },
            "description": "This suppression rule omits Lambda instance findings",
            "name": "ExampleSuppressionRuleLambda",
            "ownerId": "o-EXAMPLE222",
            "reason": "Updating filter to omit Lambda instance findings instead of
 ECR instance findings",
            "tags": {},
            "updatedAt": "2024-05-15T22:23:13.665000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des résultats dans Amazon Inspector dans le guide de l'utilisateur d'Amazon Inspector.

- Pour plus de détails sur l'API, reportez-vous UpdateFilterà la section Référence des AWS CLI commandes.

AWS IoT exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-certificate-transfer`

L'exemple de code suivant montre comment utiliser`accept-certificate-transfer`.

AWS CLI

Pour accepter un certificat d'appareil transféré depuis un autre AWS compte

L'`accept-certificate-transfer`exemple suivant accepte un certificat d'appareil transféré depuis un autre AWS compte. Le certificat est identifié par son identifiant.

```
aws iot accept-certificate-transfer \
    --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Transférer un certificat vers un autre compte dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous AcceptCertificateTransferà la section Référence des AWS CLI commandes.

`add-thing-to-billing-group`

L'exemple de code suivant montre comment utiliser`add-thing-to-billing-group`.

AWS CLI

Exemple 1 : pour ajouter un objet par son nom à un groupe de facturation

L'`add-thing-to-billing-group`exemple suivant ajoute l'objet nommé `MyLightBulb` au groupe de facturation nommé`GroupOne`.

```
aws iot add-thing-to-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyLightBulb
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour ajouter un objet par ARN à un groupe de facturation

L'`add-thing-to-billing-group`exemple suivant ajoute un objet avec un ARN spécifié à un groupe de facturation avec l'ARN spécifié. La spécification d'un ARN est utile si vous travaillez avec plusieurs AWS régions ou comptes. Cela peut vous aider à vous assurer que vous ajoutez à la bonne région et au bon compte.

```
aws iot add-thing-to-thing-group \
    --billing-group-arn "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne" \
    --thing-arn "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AddThingToBillingGroupà la section Référence des AWS CLI commandes.

`add-thing-to-thing-group`

L'exemple de code suivant montre comment utiliser`add-thing-to-thing-group`.

AWS CLI

Pour ajouter un objet à un groupe

L'`add-thing-to-thing-group`exemple suivant ajoute l'objet spécifié au groupe d'objets spécifié.

```
aws iot add-thing-to-thing-group \
    --thing-name MyLightBulb \
    --thing-group-name LightBulbs
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AddThingToThingGroupà la section Référence des AWS CLI commandes.

`associate-targets-with-job`

L'exemple de code suivant montre comment utiliser`associate-targets-with-job`.

AWS CLI

Pour associer un groupe d'objets à une tâche continue

L'`associate-targets-with-job`exemple suivant associe le groupe d'objets spécifié à la tâche continue spécifiée.

```
aws iot associate-targets-with-job \
    --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --job-id "example-job-04"
```

Sortie :

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
```

```
    "jobId": "example-job-04",
    "description": "example continuous job"
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AssociateTargetsWithJobà la section Référence des AWS CLI commandes.

`attach-policy`

L'exemple de code suivant montre comment utiliser`attach-policy`.

AWS CLI

Exemple 1 : pour associer une politique à un groupe d'objets

L'`attach-policy`exemple suivant attache la politique spécifiée à un groupe d'objets identifié par son ARN.

```
aws iot attach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "UpdateDeviceCertPolicy"
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

Exemple 2 : pour associer une politique à un certificat

L'`attach-policy`exemple suivant attache la politique `UpdateDeviceCertPolicy` au principal spécifié par un certificat.

```
aws iot attach-policy \
    --policy-name UpdateDeviceCertPolicy \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, voir Attacher une politique AWS IoT à un certificat d'appareil dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AttachPolicyà la section Référence des AWS CLI commandes.

`attach-security-profile`

L'exemple de code suivant montre comment utiliser`attach-security-profile`.

AWS CLI

Pour associer un profil de sécurité à tous les appareils non enregistrés

L'`attach-security-profile`exemple suivant associe le profil de sécurité AWS IoT Device Defender nommé `Testprofile` à tous les appareils non enregistrés de la `us-west-2` région pour ce AWS compte.

```
aws iot attach-security-profile \
    --security-profile-name Testprofile \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
unregistered-things"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AttachSecurityProfileà la section Référence des AWS CLI commandes.

`attach-thing-principal`

L'exemple de code suivant montre comment utiliser`attach-thing-principal`.

AWS CLI

Pour joindre un certificat à votre objet

L'`attach-thing-principal`exemple suivant attache un certificat à l' MyTemperatureSensor objet. Le certificat est identifié par un ARN. Vous trouverez l'ARN d'un certificat dans la console AWS IoT.

```
aws iot attach-thing-principal \
    --thing-name MyTemperatureSensor \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a351597670552 78e8
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Comment gérer les objets avec le registre dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AttachThingPrincipalà la section Référence des AWS CLI commandes.

`cancel-audit-mitigation-actions-task`

L'exemple de code suivant montre comment utiliser`cancel-audit-mitigation-actions-task`.

AWS CLI

Pour annuler une tâche d'audit relative aux mesures d'atténuation

L'`cancel-audit-mitigations-action-task`exemple suivant annule l'application des mesures d'atténuation pour la tâche spécifiée. Vous ne pouvez pas annuler des tâches déjà terminées.

```
aws iot cancel-audit-mitigation-actions-task
    --task-id "myActionsTaskId"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir CancelAuditMitigationActionsTask (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CancelAuditMitigationActionsTaskà la section Référence des AWS CLI commandes.

`cancel-audit-task`

L'exemple de code suivant montre comment utiliser`cancel-audit-task`.

AWS CLI

Pour annuler une tâche d'audit

L'`cancel-audit-task`exemple suivant annule une tâche d'audit avec l'ID de tâche spécifié. Vous ne pouvez pas annuler une tâche terminée.

```
aws iot cancel-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CancelAuditTaskà la section Référence des AWS CLI commandes.

`cancel-certificate-transfer`

L'exemple de code suivant montre comment utiliser`cancel-certificate-transfer`.

AWS CLI

Pour annuler le transfert d'un certificat vers un autre AWS compte

L'`cancel-certificate-transfer`exemple suivant annule le transfert du certificat spécifié. Le certificat est identifié par un ID de certificat. Vous pouvez trouver l'identifiant d'un certificat dans la console AWS IoT.

```
aws iot cancel-certificate-transfer \
    --certificate-
id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Transférer un certificat vers un autre compte dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous CancelCertificateTransferà la section Référence des AWS CLI commandes.

`cancel-job-execution`

L'exemple de code suivant montre comment utiliser`cancel-job-execution`.

AWS CLI

Pour annuler l'exécution d'une tâche sur un appareil

L'`cancel-job-execution`exemple suivant annule l'exécution de la tâche spécifiée sur un appareil. Si la tâche n'est pas dans `QUEUED` cet état, vous devez ajouter le `--force` paramètre.

```
aws iot cancel-job-execution \
    --job-id "example-job-03" \
    --thing-name "MyRPi"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CancelJobExecutionà la section Référence des AWS CLI commandes.

`cancel-job`

L'exemple de code suivant montre comment utiliser`cancel-job`.

AWS CLI

Pour annuler une offre d'emploi

L'`cancel-job`exemple suivant annule la tâche spécifiée.

```
aws iot cancel-job \
    --job-job "example-job-03"
```

Sortie :

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",
    "jobId": "example-job-03",
    "description": "example job test"
```

```
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CancelJobà la section Référence des AWS CLI commandes.

`clear-default-authorizer`

L'exemple de code suivant montre comment utiliser`clear-default-authorizer`.

AWS CLI

Pour effacer l'autorisateur par défaut

L'`clear-default-authorizer`exemple suivant efface l'autorisateur personnalisé par défaut actuellement configuré. Après avoir exécuté cette commande, il n'existe aucun autorisateur par défaut. Lorsque vous utilisez un autorisateur personnalisé, vous devez le spécifier par son nom dans les en-têtes de requête HTTP.

```
aws iot clear-default-authorizer
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez ClearDefaultAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ClearDefaultAuthorizerà la section Référence des AWS CLI commandes.

`confirm-topic-rule-destination`

L'exemple de code suivant montre comment utiliser`confirm-topic-rule-destination`.

AWS CLI

Pour confirmer la destination d'une règle de sujet

L'`confirm-topic-rule-destination`exemple suivant confirme la destination d'une règle de sujet à l'aide d'un jeton de confirmation reçu sur un point de terminaison HTTP.

```
aws iot confirm-topic-rule-destination \
    --confirmation-token "AYADeIcmtq-
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREFxY1E0UmlGeDg0V21BZWZ1VjZtZWFRVUJJUkt
aywpPqg8YEsa1lD4B40aJ2s1wEHKMybiF1RoOZzYisI0IvslzQY5UmCkqq3tV-3f7-
nKfosgIAAAADAAAEAAAAAAAAAAAAAAAAAAAi9RMgy-
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gTO6n50NLJVpzyQR1UmPIj5sShqXEQGc0
iufgrzTePl8RZYOWr0O6Aj9DiVzJZx-1iD6Pu-
G6PUw1kaO7Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseROlAn
PIetJ803Z4IlIlF8xXlcdPGP-PV1dOXFemyL8g"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Confirmation de la destination d'une règle de sujet dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ConfirmTopicRuleDestinationà la section Référence des AWS CLI commandes.

`create-audit-suppression`

L'exemple de code suivant montre comment utiliser`create-audit-suppression`.

AWS CLI

Pour créer une suppression des résultats d'audit

L'`create-audit-suppression`exemple suivant crée une suppression des résultats d'audit pour une politique nommée « virtualMachinePolicy » qui a été signalée comme étant trop permissive.

```
aws iot create-audit-suppression \
    --check-name IOT_POLICY_OVERLY_PERMISSIVE_CHECK \
    --resource-identifier
 policyVersionIdentifier={"policyName"="virtualMachinePolicy","policyVersionId"="1"}
 \
    --no-suppress-indefinitely \
    --expiration-date 2020-10-20
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Audit sur les suppressions dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateAuditSuppressionà la section Référence des AWS CLI commandes.

`create-authorizer`

L'exemple de code suivant montre comment utiliser`create-authorizer`.

AWS CLI

Pour créer un autorisateur personnalisé

L'`create-authorizer`exemple suivant crée un système d'autorisation personnalisé qui utilise la fonction Lambda spécifiée dans le cadre d'un service d'authentification personnalisé.

```
   aws iot create-authorizer \
       --authorizer-name "CustomAuthorizer" \
       --authorizer-function-arn "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction" \
       --token-key-name "MyAuthToken" \
       --status ACTIVE \
       --token-signing-public-keys FIRST_KEY="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
Z+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb5O3kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTAgTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
zwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa1OfL8flDB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx
50g2LW2Oy9R/dmqtDmJiVP97Z4GykxPvwlYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
lQIDAQAB
-----END PUBLIC KEY-----"
```

Sortie :

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

Pour plus d'informations, consultez CreateAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateAuthorizerà la section Référence des AWS CLI commandes.

`create-billing-group`

L'exemple de code suivant montre comment utiliser`create-billing-group`.

AWS CLI

Pour créer un groupe de facturation

L'`create-billing-group`exemple suivant crée un groupe de facturation simple nomméGroupOne.

```
aws iot create-billing-group \
    --billing-group-name GroupOne
```

Sortie :

```
{
    "billingGroupName": "GroupOne",
    "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
    "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"
}
```

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateBillingGroupà la section Référence des AWS CLI commandes.

`create-certificate-from-csr`

L'exemple de code suivant montre comment utiliser`create-certificate-from-csr`.

AWS CLI

Pour créer un certificat d'appareil à partir d'une demande de signature de certificat (CSR)

L'`create-certificate-from-csr`exemple suivant crée un certificat d'appareil à partir d'un CSR. Vous pouvez utiliser la `openssl` commande pour créer un CSR.

```
aws iot create-certificate-from-csr \
```

```
    --certificate-signing-request=file://certificate.csr
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9",
        "certificateId":
 "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9",
        "certificatePem": "<certificate-text>"
}
```

Pour plus d'informations, consultez la section CreateCertificateFromCSR dans le manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateCertificateFromCsrà la section Référence des AWS CLI commandes.

`create-custom-metric`

L'exemple de code suivant montre comment utiliser`create-custom-metric`.

AWS CLI

Pour créer une métrique personnalisée publiée par vos appareils sur Device Defender

L'`create-custom-metric`exemple suivant crée une métrique personnalisée qui mesure le pourcentage de batterie.

```
aws iot create-custom-metric \
    --metric-name "batteryPercentage" \
    --metric-type "number" \
    --display-name "Remaining battery percentage." \
    --region us-east-1 \
    --client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0"
```

Sortie :

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage"
```

```
}
```

Pour plus d'informations, consultez la section Mesures personnalisées dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous CreateCustomMetricà la section Référence des AWS CLI commandes.

`create-dimension`

L'exemple de code suivant montre comment utiliser`create-dimension`.

AWS CLI

Pour créer une dimension

Ce qui suit `create-dimension` crée une dimension avec un filtre de rubrique unique appelé`TopicFilterForAuthMessages`.

```
aws iot create-dimension \
    --name TopicFilterForAuthMessages \
    --type TOPIC_FILTER \
    --string-values device/+/auth
```

Sortie :

```
{
    "name": "TopicFilterForAuthMessages",
    "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateDimensionà la section Référence des AWS CLI commandes.

`create-domain-configuration`

L'exemple de code suivant montre comment utiliser`create-domain-configuration`.

AWS CLI

Pour créer une configuration de domaine

L'`create-domain-configuration`exemple suivant crée une configuration de domaine AWS géré avec un type de service de`DATA`.

```
aws iot create-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --service-type "DATA"
```

Sortie :

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

Pour plus d'informations, consultez la section Points de terminaison configurables dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateDomainConfigurationà la section Référence des AWS CLI commandes.

`create-dynamic-thing-group`

L'exemple de code suivant montre comment utiliser`create-dynamic-thing-group`.

AWS CLI

Pour créer un groupe d'objets dynamique

L'`create-dynamic-thing-group`exemple suivant crée un groupe d'objets dynamique qui contient tout objet dont l'attribut de température est supérieur à 60 degrés. Vous devez activer AWS l'indexation des flottes IoT avant de pouvoir utiliser des groupes d'objets dynamiques.

```
aws iot create-dynamic-thing-group \
    --thing-group-name "RoomTooWarm" \
    --query-string "attributes.temperature>60"
```

Sortie :

```
{
    "thingGroupName": "RoomTooWarm",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",
    "thingGroupId": "9d52492a-fc87-43f4-b6e2-e571d2ffcad1",
    "indexName": "AWS_Things",
    "queryString": "attributes.temperature>60",
    "queryVersion": "2017-09-30"
}
```

Pour plus d'informations, consultez la section Dynamic Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateDynamicThingGroupà la section Référence des AWS CLI commandes.

`create-job`

L'exemple de code suivant montre comment utiliser`create-job`.

AWS CLI

Exemple 1 : pour créer une tâche

L'`create-job`exemple suivant crée une tâche AWS IoT simple qui envoie un document JSON à l'`MyRaspberryPi`appareil.

```
aws iot create-job \
    --job-id "example-job-01" \
    --targets "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi" \
    --document file://example-job.json \
    --description "example job test" \
    --target-selection SNAPSHOT
```

Sortie :

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
    "jobId": "example-job-01",
    "description": "example job test"
```

```
}
```

Exemple 2 : pour créer une tâche continue

L'`create-job`exemple suivant crée une tâche qui continue de s'exécuter une fois que les éléments spécifiés comme cibles ont terminé la tâche. Dans cet exemple, la cible est un groupe d'objets. Ainsi, lorsque de nouveaux appareils sont ajoutés au groupe, le travail continu s'exécute sur ces nouveaux objets.

aws iot create-job --job-id « example-job-04" --targets « arn:aws:iot:us-west- 2:123456789012 : -job.json --description « exemple de tâche continue » --target-selection CONTINUOUS thinggroup/DeadBulbs" --document file://example

Sortie :

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
    "jobId": "example-job-04",
    "description": "example continuous job"
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateJobà la section Référence des AWS CLI commandes.

`create-keys-and-certificate`

L'exemple de code suivant montre comment utiliser`create-keys-and-certificate`.

AWS CLI

Pour créer une paire de clés RSA et émettre un certificat X.509

Ce qui suit `create-keys-and-certificate` crée une paire de clés RSA 2048 bits et émet un certificat X.509 à l'aide de la clé publique émise. Comme c'est la seule fois où l' AWS IoT fournit la clé privée pour ce certificat, veillez à le conserver dans un endroit sécurisé.

```
aws iot create-keys-and-certificate \
    --certificate-pem-outfile "myTest.cert.pem" \
```

```
    --public-key-outfile "myTest.public.key" \
    --private-key-outfile "myTest.private.key"
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificateId":
 "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSEXAMPLE2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEXAMPLE25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZEXAMPLELG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkEXAMPLEQEFAAOCAQ8AMIIBCgKCAQEAEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEuuN/
dMAS3fyce8DW/4+EXAMPLEyjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqXOgHh/
xJTwo
+sGpWEXAMPLEDz18xOd2ka4tCzuWEXAMPLEahJbYkCPUBSU8opVkR7qkEXAMPLE1DR6sx2HocliOOLtu6Fkw91swQWEX
\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEvp9mQOUXP6plfgxwKRX2fEXAMPLEDa
\nhJLXkX3rHU2xbxJSq7D+XEXAMPLEcw+LyFhI5mgFRl88eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----
END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omittted for security
 reasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

Pour plus d'informations, consultez la section Création et enregistrement d'un certificat d'appareil AWS IoT dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateKeysAndCertificateà la section Référence des AWS CLI commandes.

`create-mitigation-action`

L'exemple de code suivant montre comment utiliser`create-mitigation-action`.

AWS CLI

Pour créer une action d'atténuation

L'`create-mitigation-action`exemple suivant définit une action d'atténuation nommée `AddThingsToQuarantineGroup1Action` qui, lorsqu'elle est appliquée, déplace les objets dans le groupe d'objets nommé`QuarantineGroup1`. Cette action remplace les groupes d'objets dynamiques.

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

Contenu de `params.json` :

```
{
    "actionName": "AddThingsToQuarantineGroup1Action",
    "actionParams": {
        "addThingsToThingGroupParams": {
            "thingGroupNames": [
                "QuarantineGroup1"
            ],
            "overrideDynamicGroups": true
        }
    },
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole"
}
```

Sortie :

```
{
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroup1Action",
    "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
```

```
}
```

Pour plus d'informations, voir CreateMitigationAction (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateMitigationActionà la section Référence des AWS CLI commandes.

`create-ota-update`

L'exemple de code suivant montre comment utiliser`create-ota-update`.

AWS CLI

Pour créer une mise à jour OTA à utiliser avec Amazon FreeRTOS

L'`create-ota-update`exemple suivant crée un AWS IoT OTAUpdate sur un groupe cible d'objets ou de groupes. Cela fait partie d'une mise à jour d'Amazon over-the-air FreeRTOS qui vous permet de déployer de nouvelles images de microprogramme sur un seul appareil ou un groupe d'appareils.

```
aws iot create-ota-update \
    --cli-input-json file://create-ota-update.json
```

Contenu de `create-ota-update.json` :

```
{
    "otaUpdateId": "ota12345",
    "description": "A critical update needed right away.",
    "targets": [
        "device1",
        "device2",
        "device3",
        "device4"
    ],
    "targetSelection": "SNAPSHOT",
    "awsJobExecutionsRolloutConfig": {
        "maximumPerMinute": 10
    },
    "files": [
        {
```

```
            "fileName": "firmware.bin",
            "fileLocation": {
              "stream": {
                "streamId": "004",
                "fileId":123
              }
            },
            "codeSigning": {
              "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
            }
          }
    ]
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
}
```

Sortie :

```
{
     "otaUpdateId": "ota12345",
     "awsIotJobId": "job54321",
     "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
     "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/itsajob",
     "otaUpdateStatus": "CREATE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez la section Créer OTAUpdate dans le guide de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateOtaUpdateà la section Référence des AWS CLI commandes.

`create-policy-version`

L'exemple de code suivant montre comment utiliser`create-policy-version`.

AWS CLI

Pour mettre à jour une politique avec une nouvelle version

L'`create-policy-version`exemple suivant met à jour une définition de stratégie en créant une nouvelle version de stratégie. Cet exemple fait également de la nouvelle version la version par défaut.

```
aws iot create-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-document file://policy.json \
    --set-as-default
```

Contenu de `policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action":  "iot:UpdateCertificate",
            "Resource": "*"
        }
    ]
}
```

Sortie :

```
{
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
    "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
 \"Allow\", \"Action\":  \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
    "policyVersionId": "2",
    "isDefaultVersion": true
}
```

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyVersionà la section Référence des AWS CLI commandes.

`create-policy`

L'exemple de code suivant montre comment utiliser`create-policy`.

AWS CLI

Pour créer une politique AWS IoT

L'`create-policy`exemple suivant crée une politique AWS IoT nommée TemperatureSensorPolicy. Le `policy.json` fichier contient des déclarations qui autorisent les actions politiques AWS relatives à l'IoT.

```
aws iot create-policy \
    --policy-name TemperatureSensorPolicy \
    --policy-document file://policy.json
```

Contenu de `policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:123456789012:topic/topic_1",
                "arn:aws:iot:us-west-2:123456789012:topic/topic_2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1",
                "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:123456789012:client/basicPubSub"
            ]
        }
```

```
    ]
}
```

Sortie :

```
{
    "policyName": "TemperatureSensorPolicy",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
    "policyDocument": "{
        \"Version\": \"2012-10-17\",
        \"Statement\": [
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Publish\",
                    \"iot:Receive\"
                ],
                \"Resource\": [
                    \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
                    \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
                ]
            },
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Subscribe\"
                ],
                \"Resource\": [
                    \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1\",
                    \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2\"
                ]
            },
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Connect\"
                ],
                \"Resource\": [
                    \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub\"
                ]
            }
        ]
```

```
    }",
    "policyVersionId": "1"
}
```

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyà la section Référence des AWS CLI commandes.

`create-provisioning-claim`

L'exemple de code suivant montre comment utiliser`create-provisioning-claim`.

AWS CLI

Pour créer une demande de provisionnement

L'`create-provisioning-claim`exemple suivant crée une demande d'approvisionnement à partir d'un modèle de provisionnement.

```
aws iot create-provisioning-claim \
    --template-name MyTestProvisioningTemplate
```

Sortie :

```
{
    "certificateId":
 "78de02184b2ce80cf8fb709bda59e62b19fb83513590483eb0434589476ab09f",
    "certificatePem": "-----BEGIN CERTIFICATE-----\nMIIDdzCCAl
+gAwIBAgIUXSZhEBLztMLZ2fHG
14gV0NymYY0wDQYJKoZIhvcNAQEL
\nBQAwfjELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBg
VBAcM\nB1NlYXR0bGUxGDAWBgNVBAoMD0FtYXpvbi5jb20gSW5jLjEgMB4GA1UECwwXQW1h
\nem9uIElvVCBQcm9
2aXNpb25pbmcxDDAKBgNVBAUTAzEuMDAeFw0yMDA3MjgxNjQ0\nMDZaFw0yMDA3MjgxNjUxMDZaMEsxSTBHBgNVB
AMMQDFhNDEyM2VkNmIxYjU3MzE3\nZTgzMTJmY2MzN2FiNTdhY2MzYTZkZGVjOGQ5OGY3NzUwMWRlMjc0YjhmYTQ
xN2Iw\nggEiMA0GCSqGSIb3EXAMPLEAA4IBDwAwggEKAoIBAQDBhKI94ktKLqTwnj+ayOq1\nTAJt/
N6s6IJDZvl
rYjkC0E7wzaeY3TprWk03S29vUzVuEOXHXQXZbihgpg2m6fza\nkWm9/
wpjzE9ny5+xkPGVH4Wnwz7yK5m8S0agL
T96cRBSWnWmonOWdY0GKVzni0CA\n+iyGudgrFKm7Eae/
v18oXrf82KtOAGO4xG0KE2WKYHsT1fx3c9xZhlXP/eX
```

```
Lhv00\n+lGp0WVw9PbhKfrxliKJ5q6sL5nVUaUHq6hlQPYwsATeOvAp3u0ak5zgTyL0fg7Y
\nPyKk6VYwLW62r+V
YBSForEMOAhkq3LsP/rjxpEKmi2W4lPVS6oFZRKcD+H1Kyil5\nAgMBAAGjIDAeMAwGA1UdEwEB/
wQCMAAwDgYDV
R0PAQH/BAQDAgeAMA0GCSqGSIb3\nDQEBCwUAA4IBAQAGgix2k6nVqbZFKq97/fZBzLGS0dyz5rT/
E41cDIRX+1j
EPW41\nw0D+2sXheCZLZZnSkvIiP74IToNeXDrjdcaodeGFVHIElRjhMIq+4ZebPbRLtidF
\nRc2hfcTAlqq9Z6v
5Vk6BeM1tu0RqH1wPoVUccLPya8EjNCbnJZUmGdOfrN/Y9pho\n5ikV+HPeZhG/k6dhE2GsQJyKFVHL/
uBgKSily
1bRyWU1r6qcpWBNBHjUoD7HgOwD
\nnzMh4XRb2FQDsqFalkCSYmeL8IVC49sgPD9Otyp5uteGMTy62usAAUQdq/f
ZvrWg\nOkFpwMVnGKVKT7Kg0kKOLzKWOBB2Jm4/gmrJ\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCg
KCAQEAwYSiPeJLSi6k8J4/msjq
\ntUwCbfzerOiCQ2b5a2I5AtBO8M2nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYN
pun8\n2pFpvf8KY8xPZ8ufsZDxlR+Fp8M+8iuZvEtGoC0/enEQUlp1pqJzlnWNBilc54tA
\ngPoshrnYKxSpuxGn
v79fKF63/NirTgBjuMRtChNlimEXAMPLE3PcWYZVz/3ly4b9\nNPpRqdFlcPT24Sn68ZYiieaurC
+Z1VGlB6uoZU
D2MLAE3jrwKd7tGpOc4E8i9H4O\n2D8ipOlWMC1utq/
lWAUhaKxDDgIZKty7D/648aRCpotluJT1UuqBWUSnA/h9
Ssop\neQIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAwYSiPeJLSi6k8J4/
msjqtUwCbfzerOiCQ2b5a2I5AtBO8M2n
\nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYNpun82pFpvf8KY8xPZ8ufsZ
DxlR+Fp8M+\n8iuZvEtGoC0/enEQUlp1pqJzlnWNBilc54tAgPoshrnYKxSpuxGnv79fKF63/Nir
\nTgBjuMRtCh
NlimB7E9X8d3PcWYZVz/3ly4b9NPpRqdFlcPT24Sn68ZYiieaurC+Z
\n1VGlB6uoZUD2MLAE3jrwKd7tGpOc4E8i
9H4O2D8ipOlWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABAoIBAEAybN
QUtx9T2/nK\ntZT2pA4iugecxI4dz+DmT0XVXs5VJmrx/
nBSq6ejXExEpSIMO4RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1krOIPuxWebNOrRoPZr+1RSer+wv2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9FdOUoQxdjHBHaM5PDmJ
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nou0+91eTOP+/QfP7P8ZjOIkO2XivlRcVDyN/
E4QXPKuIkM/8vS8VK+
E9pATQ0MtB\n2lw8R/YU5AJd6jlEXAMPLEGU2UzRzInNWiLtkPPPqgqXXhxOf+mxByjcMalVJk0L
\nhOG2ROUCgY
```

```
EA+ROcHNHy/XbsP7FihOhEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5SlMBp7Slidxak
bIn28fKfPn62DaemGCIoyDgLpF+eUxBx
\ngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAxuWt\npzdWWmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjsks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDXuxDe+V369Bianx8aZFyIsckA7OwXW1w1h
\ngRC5rQ4XOgp3+Jmw7eAO8LRYDjaN846+
QbtO2KcCgYAWS0UL51bijQR0ZwI0dz27\nFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
BRtASWODab\naHXsDhVm2Jye+ESLoHMaCLoyCkT3ll8yqXIcEDStMO7fO1Ryag164EiJvSIrMfny\nNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2xONZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXghOeoTpC8VXkzcnzouTiQ
\neFQQSGfnp7Ioe6UIz23715pKduzSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nPRTIZQKBgA8MASXtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJCkAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nvdlc5cFQewWb4/
zqAtPdinkPlN94ileI
79XQdc7RlJ0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
    },
    "expiration": 1595955066.0
}
```

Pour plus d'informations, consultez la section Provisionnement par un utilisateur de confiance dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous CreateProvisioningClaimà la section Référence des AWS CLI commandes.

`create-provisioning-template-version`

L'exemple de code suivant montre comment utiliser`create-provisioning-template-version`.

AWS CLI

Pour créer une version du modèle de provisionnement

L'exemple suivant crée une version pour le modèle de provisionnement spécifié. Le corps de la nouvelle version est fourni dans le fichier`template.json`.

```
aws iot create-provisioning-template-version \
```

```
    --template-name widget-template \
    --template-body file://template.json
```

Contenu de `template.json` :

```
{
    "Parameters" : {
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["widgets", "WA"],
                "BillingGroup": "BillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
```

```
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/foo/
bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
 "LocationUrl"]}
        }
    }
}
```

Sortie :

```
{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "versionId": 2,
    "isDefaultVersion": false
}
```

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous CreateProvisioningTemplateVersionà la section Référence des AWS CLI commandes.

`create-provisioning-template`

L'exemple de code suivant montre comment utiliser`create-provisioning-template`.

AWS CLI

Pour créer un modèle de provisionnement

L'`create-provisioning-template`exemple suivant crée un modèle de provisionnement tel que défini par le fichier`template.json`.

```
aws iot create-provisioning-template \
    --template-name widget-template \
    --description "A provisioning template for widgets" \
    --provisioning-role-arn arn:aws:iam::123456789012:role/Provision_role \
    --template-body file://template.json
```

Contenu de `template.json` :

```
{
    "Parameters" : {
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
```

```
                "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["widgets", "WA"],
                "BillingGroup": "BillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:504350838278:topic/foo/
bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
 "LocationUrl"]}
        }
    }
}
```

Sortie :

```
{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "defaultVersionId": 1
}
```

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous CreateProvisioningTemplateà la section Référence des AWS CLI commandes.

`create-role-alias`

L'exemple de code suivant montre comment utiliser`create-role-alias`.

AWS CLI

Pour créer un alias de rôle

L'`create-role-alias`exemple suivant crée un alias de rôle appelé `LightBulbRole` pour le rôle spécifié.

```
aws iot create-role-alias \
    --role-alias LightBulbRole \
    --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Sortie :

```
{
    "roleAlias": "LightBulbRole",
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

Pour plus d'informations, consultez CreateRoleAliasle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateRoleAliasà la section Référence des AWS CLI commandes.

`create-scheduled-audit`

L'exemple de code suivant montre comment utiliser`create-scheduled-audit`.

AWS CLI

Pour créer un audit planifié

L'`create-scheduled-audit`exemple suivant crée un audit planifié qui s'exécute chaque semaine, le mercredi, pour vérifier si les certificats CA ou les certificats d'appareil expirent.

```
aws iot create-scheduled-audit \
    --scheduled-audit-name WednesdayCertCheck \
    --frequency WEEKLY \
    --day-of-week WED \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK
```

Sortie :

```
{
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateScheduledAudità la section Référence des AWS CLI commandes.

`create-security-profile`

L'exemple de code suivant montre comment utiliser`create-security-profile`.

AWS CLI

Pour créer un profil de sécurité

L'`create-security-profile`exemple suivant crée un profil de sécurité qui vérifie si la bande passante cellulaire dépasse un seuil ou si plus de 10 échecs d'autorisation se produisent sur une période de cinq minutes.

```
aws iot create-security-profile \
    --security-profile-name PossibleIssue \
    --security-profile-description "Check to see if authorization fails 10 times in
 5 minutes or if cellular bandwidth exceeds 128"  \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":10},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

Sortie :

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue"
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateSecurityProfileà la section Référence des AWS CLI commandes.

`create-stream`

L'exemple de code suivant montre comment utiliser`create-stream`.

AWS CLI

Pour créer un flux permettant de distribuer un ou plusieurs fichiers volumineux en morceaux via MQTT

L'`create-stream`exemple suivant crée un flux pour distribuer un ou plusieurs fichiers volumineux par blocs via MQTT. Un flux transfère des octets de données dans des fragments ou blocs packagés sous forme de messages MQTT à partir d'une source telle que S3. Un ou plusieurs fichiers peuvent être associés à un flux.

```
aws iot create-stream \
    --cli-input-json file://create-stream.json
```

Contenu de `create-stream.json` :

```
{
    "streamId": "stream12345",
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
    "files": [
        {
            "fileId": 123,
            "s3Location": {
                "bucket":"codesign-ota-bucket",
                "key":"48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
            }
        }
    ],
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"
}
```

Sortie :

```
{
     "streamId": "stream12345",
     "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
     "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
     "streamVersion": "1"
}
```

Pour plus d'informations, consultez CreateStreamle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateStreamà la section Référence des AWS CLI commandes.

`create-thing-group`

L'exemple de code suivant montre comment utiliser`create-thing-group`.

AWS CLI

Exemple 1 : pour créer un groupe d'objets

L'`create-thing-group`exemple suivant crée un groupe d'objets nommé `LightBulbs` avec une description et deux attributs.

```
aws iot create-thing-group \
    --thing-group-name LightBulbs \
    --thing-group-properties "thingGroupDescription=\"Generic bulb group\",
 attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

Sortie :

```
{
    "thingGroupName": "LightBulbs",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",
    "thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"
}
```

Exemple 2 : pour créer un groupe d'objets faisant partie d'un groupe parent

Ce qui suit `create-thing-group` crée un groupe d'objets nommé `HalogenBulbs` dont le groupe d'objets parent est nommé`LightBulbs`.

```
aws iot create-thing-group \
    --thing-group-name HalogenBulbs \
    --parent-group-name LightBulbs
```

Sortie :

```
{
    "thingGroupName": "HalogenBulbs",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",
    "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateThingGroupà la section Référence des AWS CLI commandes.

`create-thing-type`

L'exemple de code suivant montre comment utiliser`create-thing-type`.

AWS CLI

Pour définir un type d'objet

L'`create-thing-type`exemple suivant définit un type d'objet et les attributs associés.

```
aws iot create-thing-type \
    --thing-type-name "LightBulb" \
    --thing-type-properties "thingTypeDescription=light bulb type,
 searchableAttributes=wattage,model"
```

Sortie :

```
{
    "thingTypeName": "LightBulb",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"
}
```

Pour plus d'informations, consultez la section Types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateThingTypeà la section Référence des AWS CLI commandes.

`create-thing`

L'exemple de code suivant montre comment utiliser`create-thing`.

AWS CLI

Exemple 1 : pour créer un enregistrement d'objet dans le registre

L'`create-thing`exemple suivant crée une entrée pour un appareil dans le registre AWS des objets IoT.

```
aws iot create-thing \
    --thing-name SampleIoTThing
```

Sortie :

```
{
```

```
    "thingName": "SampleIoTThing",
    "thingArn": "arn:aws:iot:us-west-2: 123456789012:thing/SampleIoTThing",
    "thingId": " EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "
}
```

Exemple 2 : pour définir un objet associé à un type d'objet

L'`create-thing`exemple suivant crée un objet doté du type d'objet spécifié et de ses attributs.

```
aws iot create-thing \
    --thing-name "MyLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

Sortie :

```
{
    "thingName": "MyLightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

Pour plus d'informations, consultez la section Comment gérer les objets avec le registre et les types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateThingà la section Référence des AWS CLI commandes.

`create-topic-rule-destination`

L'exemple de code suivant montre comment utiliser`create-topic-rule-destination`.

AWS CLI

Pour créer une destination de règle de sujet

L'`create-topic-rule-destination`exemple suivant crée une destination de règle de sujet pour un point de terminaison HTTP.

```
aws iot create-topic-rule-destination \
    --destination-configuration httpUrlConfiguration={confirmationUrl=https://
example.com}
```

Sortie :

```
{
    "topicRuleDestination": {
        "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "status": "IN_PROGRESS",
        "statusReason": "Awaiting confirmation. Confirmation message sent on
 2020-07-09T22:47:54.154Z; no response received from the endpoint.",
        "httpUrlProperties": {
            "confirmationUrl": "https://example.com"
        }
    }
}
```

Pour plus d'informations, consultez la section Création d'une destination de règle de sujet dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateTopicRuleDestinationà la section Référence des AWS CLI commandes.

`create-topic-rule`

L'exemple de code suivant montre comment utiliser`create-topic-rule`.

AWS CLI

Pour créer une règle qui envoie une alerte Amazon SNS

L'`create-topic-rule`exemple suivant crée une règle qui envoie un message Amazon SNS lorsque les relevés du niveau d'humidité du sol, tels qu'ils apparaissent dans l'ombre d'un appareil, sont faibles.

```
aws iot create-topic-rule \
    --rule-name "LowMoistureRule" \
    --topic-rule-payload file://plant-rule.json
```

L'exemple nécessite que le code JSON suivant soit enregistré dans un fichier nommé `plant-rule.json` :

```
{
```

```
    "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
 state.reported.moisture = 'low'\n",
    "description": "Sends an alert whenever soil moisture level readings are too
 low.",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [{
            "sns": {
                "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
                "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
                "messageFormat": "RAW"
            }
    }]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une règle AWS IoT dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateTopicRuleà la section Référence des AWS CLI commandes.

`delete-account-audit-configuration`

L'exemple de code suivant montre comment utiliser`delete-account-audit-configuration`.

AWS CLI

Pour désactiver toutes les vérifications d'audit de votre AWS compte

L'`delete-account-audit-configuration`exemple suivant restaure les paramètres par défaut d' AWS IoT Device Defender pour ce compte, en désactivant tous les contrôles d'audit et en effaçant les données de configuration. Il supprime également tous les audits planifiés pour ce compte. Utilisez cette commande avec prudence.

```
aws iot delete-account-audit-configuration \
    --delete-scheduled-audits
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteAccountAuditConfigurationà la section Référence des AWS CLI commandes.

`delete-audit-suppression`

L'exemple de code suivant montre comment utiliser`delete-audit-suppression`.

AWS CLI

Pour supprimer une suppression des résultats d'audit

L'`delete-audit-suppression`exemple suivant supprime une suppression des résultats d'audit pour DEVICE_CERTIFICATE_EXPIRING_CHECK.

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Audit sur les suppressions dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteAuditSuppressionà la section Référence des AWS CLI commandes.

`delete-authorizer`

L'exemple de code suivant montre comment utiliser`delete-authorizer`.

AWS CLI

Pour supprimer un autorisateur personnalisé

L'`delete-authorizer`exemple suivant supprime l'autorisateur nommé. `CustomAuthorizer` Un autorisateur personnalisé doit être dans `INACTIVE` cet état pour que vous puissiez le supprimer.

```
aws iot delete-authorizer \
```

```
    --authorizer-name CustomAuthorizer
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteAuthorizerle Guide du développeur de AWS l'IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteAuthorizerà la section Référence des AWS CLI commandes.

`delete-billing-group`

L'exemple de code suivant montre comment utiliser`delete-billing-group`.

AWS CLI

Pour supprimer un groupe de facturation

L'`delete-billing-group`exemple suivant supprime le groupe de facturation spécifié. Vous pouvez supprimer un groupe de facturation même s'il contient un ou plusieurs éléments.

```
aws iot delete-billing-group \
    --billing-group-name BillingGroupTwo
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteBillingGroupà la section Référence des AWS CLI commandes.

`delete-ca-certificate`

L'exemple de code suivant montre comment utiliser`delete-ca-certificate`.

AWS CLI

Pour supprimer un certificat CA

L'`delete-ca-certificate`exemple suivant supprime le certificat CA avec l'ID de certificat spécifié.

```
aws iot delete-ca-certificate \
```

```
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer CACertificate dans le manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteCaCertificateà la section Référence des AWS CLI commandes.

`delete-certificate`

L'exemple de code suivant montre comment utiliser`delete-certificate`.

AWS CLI

Pour supprimer un certificat d'appareil

L'`delete-certificate`exemple suivant supprime le certificat de l'appareil avec l'ID spécifié.

```
aws iot delete-certificate \
    --certificate-
id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteCertificatele manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteCertificateà la section Référence des AWS CLI commandes.

`delete-custom-metric`

L'exemple de code suivant montre comment utiliser`delete-custom-metric`.

AWS CLI

Pour supprimer une métrique personnalisée

L'`delete-custom-metric`exemple suivant supprime une métrique personnalisée.

```
aws iot delete-custom-metric \
```

```
    --metric-name batteryPercentage \
    --region us-east-1
```

Sortie :

```
HTTP 200
```

Pour plus d'informations, consultez la section Mesures personnalisées dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous DeleteCustomMetricà la section Référence des AWS CLI commandes.

`delete-dimension`

L'exemple de code suivant montre comment utiliser`delete-dimension`.

AWS CLI

Pour supprimer une dimension

L'`delete-dimension`exemple suivant supprime une dimension appelée`TopicFilterForAuthMessages`.

```
aws iot delete-dimension \
    --name TopicFilterForAuthMessages
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteDimensionà la section Référence des AWS CLI commandes.

`delete-domain-configuration`

L'exemple de code suivant montre comment utiliser`delete-domain-configuration`.

AWS CLI

Pour supprimer une configuration de domaine

L'`delete-domain-configuration`exemple suivant supprime une configuration de domaine nommée `additionalDataDomain` depuis votre AWS compte.

```
aws iot delete-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --domain-configuration-status "OK"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Points de terminaison configurables dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteDomainConfigurationà la section Référence des AWS CLI commandes.

`delete-dynamic-thing-group`

L'exemple de code suivant montre comment utiliser`delete-dynamic-thing-group`.

AWS CLI

Pour supprimer un groupe d'objets dynamique

L'`delete-dynamic-thing-group`exemple suivant supprime le groupe d'objets dynamiques spécifié.

```
aws iot delete-dynamic-thing-group \
    --thing-group-name "RoomTooWarm"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Dynamic Thing Groups dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteDynamicThingGroupà la section Référence des AWS CLI commandes.

`delete-job-execution`

L'exemple de code suivant montre comment utiliser`delete-job-execution`.

AWS CLI

Pour supprimer l'exécution d'une tâche

L'`delete-job-execution`exemple suivant supprime l'exécution de la tâche spécifiée sur un appareil. `describe-job-execution`À utiliser pour obtenir le numéro d'exécution.

```
aws iot delete-job-execution
    --job-id "example-job-02"
    --thing-name "MyRaspberryPi"
    --execution-number 1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteJobExecutionà la section Référence des AWS CLI commandes.

`delete-job`

L'exemple de code suivant montre comment utiliser`delete-job`.

AWS CLI

Pour supprimer une tâche

L'`delete-job`exemple suivant supprime la tâche spécifiée. En spécifiant `--force` cette option, la tâche est supprimée même si son statut est`IN_PROGRESS`.

```
aws iot delete-job \
    --job-id "example-job-04" \
    --force
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteJobà la section Référence des AWS CLI commandes.

`delete-mitigation-action`

L'exemple de code suivant montre comment utiliser`delete-mitigation-action`.

AWS CLI

Pour supprimer une action d'atténuation

L'`delete-mitigation-action`exemple suivant supprime l'action d'atténuation spécifiée.

```
aws iot delete-mitigation-action \
    --action-name AddThingsToQuarantineGroup1Action
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir DeleteMitigationAction (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteMitigationActionà la section Référence des AWS CLI commandes.

`delete-ota-update`

L'exemple de code suivant montre comment utiliser`delete-ota-update`.

AWS CLI

Pour supprimer une mise à jour OTA

L'`delete-ota-update`exemple suivant supprime la mise à jour OTA spécifiée.

```
aws iot delete-ota-update \
    --ota-update-id ota12345 \
    --delete-stream \
    --force-delete-aws-job
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Supprimer OTAUpdate dans le manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteOtaUpdateà la section Référence des AWS CLI commandes.

`delete-policy-version`

L'exemple de code suivant montre comment utiliser`delete-policy-version`.

AWS CLI

Pour supprimer une version de la politique

L'`delete-policy-version`exemple suivant supprime la version 2 de la politique spécifiée de votre AWS compte.

```
aws iot delete-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyVersionà la section Référence des AWS CLI commandes.

`delete-policy`

L'exemple de code suivant montre comment utiliser`delete-policy`.

AWS CLI

Pour supprimer une politique

L'`delete-policy`exemple suivant supprime la politique spécifiée de votre AWS compte.

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`delete-provisioning-template-version`

L'exemple de code suivant montre comment utiliser`delete-provisioning-template-version`.

AWS CLI

Pour supprimer une version d'un modèle de provisionnement

L'`delete-provisioning-template-version`exemple suivant supprime la version 2 du modèle de provisioning spécifié.

```
aws iot delete-provisioning-template-version \
    --version-id 2 \
    --template-name "widget-template"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous DeleteProvisioningTemplateVersionà la section Référence des AWS CLI commandes.

`delete-provisioning-template`

L'exemple de code suivant montre comment utiliser`delete-provisioning-template`.

AWS CLI

Pour supprimer un modèle de provisionnement

L'`delete-provisioning-template`exemple suivant supprime le modèle de provisioning spécifié.

```
aws iot delete-provisioning-template \
    --template-name widget-template
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous DeleteProvisioningTemplateà la section Référence des AWS CLI commandes.

`delete-registration-code`

L'exemple de code suivant montre comment utiliser`delete-registration-code`.

AWS CLI

Pour supprimer votre code d'enregistrement

L'`delete-registration-code`exemple suivant supprime un code d'enregistrement spécifique à un compte AWS IoT.

```
aws iot delete-registration-code
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utiliser votre propre certificat dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteRegistrationCodeà la section Référence des AWS CLI commandes.

`delete-role-alias`

L'exemple de code suivant montre comment utiliser`delete-role-alias`.

AWS CLI

Pour supprimer un alias de rôle AWS IoT

L'`delete-role-alias`exemple suivant supprime un alias de rôle AWS IoT nommé`LightBulbRole`.

```
aws iot delete-role-alias \
    --role-alias LightBulbRole
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Autorisation des appels directs vers AWS des services dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteRoleAliasà la section Référence des AWS CLI commandes.

`delete-scheduled-audit`

L'exemple de code suivant montre comment utiliser`delete-scheduled-audit`.

AWS CLI

Pour supprimer un audit planifié

L'`delete-scheduled-audit`exemple suivant supprime l'audit planifié AWS IoT Device Defender nomméAWSIoTDeviceDefenderDailyAudit.

```
aws iot delete-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteScheduledAudità la section Référence des AWS CLI commandes.

`delete-security-profile`

L'exemple de code suivant montre comment utiliser`delete-security-profile`.

AWS CLI

Pour supprimer un profil de sécurité

L'`delete-security-profile`exemple suivant supprime un profil de sécurité nomméPossibleIssue.

```
aws iot delete-security-profile \
    --security-profile-name PossibleIssue
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteSecurityProfileà la section Référence des AWS CLI commandes.

`delete-stream`

L'exemple de code suivant montre comment utiliser`delete-stream`.

AWS CLI

Pour supprimer un flux

L'`delete-stream`exemple suivant supprime le flux spécifié.

```
aws iot delete-stream \
    --stream-id stream12345
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteStreamle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteStreamà la section Référence des AWS CLI commandes.

`delete-thing-group`

L'exemple de code suivant montre comment utiliser`delete-thing-group`.

AWS CLI

Pour supprimer un groupe d'objets

L'`delete-thing-group`exemple suivant supprime le groupe d'objets spécifié. Vous ne pouvez pas supprimer un groupe d'objets s'il contient des groupes d'objets enfants.

```
aws iot delete-thing-group \
    --thing-group-name DefectiveBulbs
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteThingGroupà la section Référence des AWS CLI commandes.

`delete-thing-type`

L'exemple de code suivant montre comment utiliser`delete-thing-type`.

AWS CLI

Exemple 1 : pour supprimer un type d'objet

L'`delete-thing-type`exemple suivant supprime un type d'objet obsolète.

```
aws iot delete-thing-type \
    --thing-type-name "obsoleteThingType"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteThingTypeà la section Référence des AWS CLI commandes.

`delete-thing`

L'exemple de code suivant montre comment utiliser`delete-thing`.

AWS CLI

Pour afficher des informations détaillées sur un objet

L'`delete-thing`exemple suivant supprime un objet du registre AWS IoT de votre AWS compte.

aidez-moi à supprime-thing --thing-name « » FourthBulb

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Comment gérer les objets avec le registre dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir DeleteThingla section Référence des AWS CLI commandes.

`delete-topic-rule-destination`

L'exemple de code suivant montre comment utiliser`delete-topic-rule-destination`.

AWS CLI

Pour supprimer une destination de règle de sujet

L'`delete-topic-rule-destination`exemple suivant supprime la destination de la règle de rubrique spécifiée.

```
aws iot delete-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'une destination de règle de sujet dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DeleteTopicRuleDestinationla section Référence des AWS CLI commandes.

`delete-topic-rule`

L'exemple de code suivant montre comment utiliser`delete-topic-rule`.

AWS CLI

Pour supprimer une règle

L'`delete-topic-rule`exemple suivant supprime la règle spécifiée.

```
aws iot delete-topic-rule \
    --rule-name "LowMoistureRule"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Supprimer une règle dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DeleteTopicRulela section Référence des AWS CLI commandes.

`delete-v2-logging-level`

L'exemple de code suivant montre comment utiliser`delete-v2-logging-level`.

AWS CLI

Pour supprimer le niveau de journalisation d'un groupe d'objets

L'`delete-v2-logging-level`exemple suivant supprime le niveau de journalisation pour le groupe d'objets spécifié.

```
aws iot delete-v2-logging-level \
    --target-type THING_GROUP \
    --target-name LightBulbs
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, voir DeleteV2 LoggingLevel dans AWS CLI Command Reference.

`deprecate-thing-type`

L'exemple de code suivant montre comment utiliser`deprecate-thing-type`.

AWS CLI

Exemple 1 : Pour déprécier un type d'objet

L'`deprecate-thing-type`exemple suivant déconseille un type d'objet afin que les utilisateurs ne puissent pas y associer de nouveaux éléments.

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour annuler la dépréciation d'un type d'objet

L'`deprecate-thing-type`exemple suivant inverse la dépréciation d'un type d'objet, ce qui permet aux utilisateurs d'y associer à nouveau de nouveaux éléments.

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType" \
    --undo-deprecate
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DeprecateThingTypela section Référence des AWS CLI commandes.

`describe-account-audit-configuration`

L'exemple de code suivant montre comment utiliser`describe-account-audit-configuration`.

AWS CLI

Pour consulter les paramètres de configuration d'audit actuels

L'`describe-account-audit-configuration`exemple suivant répertorie les paramètres actuels de votre configuration d'audit AWS IoT Device Defender.

```
aws iot describe-account-audit-configuration
```

Sortie :

```
{
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTDeviceDefenderAudit_1551201085996",
    "auditNotificationTargetConfigurations": {
        "SNS": {
            "targetArn": "arn:aws:sns:us-west-2:123456789012:ddaudits",
            "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTDeviceDefenderAudit",
            "enabled": true
        }
    },
```

```
    "auditCheckConfigurations": {
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "CA_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "CONFLICTING_CLIENT_IDS_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_SHARED_CHECK": {
            "enabled": true
        },
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "LOGGING_DISABLED_CHECK": {
            "enabled": true
        },
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": true
        },
        "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": true
        },
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        }
    }
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeAccountAuditConfigurationla section Référence des AWS CLI commandes.

`describe-audit-finding`

L'exemple de code suivant montre comment utiliser`describe-audit-finding`.

AWS CLI

Pour répertorier les détails d'un résultat d'audit

L'`describe-audit-finding`exemple suivant répertorie les détails du résultat d'audit AWS IoT Device Defender spécifié. Un audit peut donner lieu à de multiples constatations. Utilisez la `list-audit-findings` commande pour obtenir la liste des résultats d'un audit afin d'obtenir le`findingId`.

```
aws iot describe-audit-finding \
    --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"
```

Sortie :

```
{
    "finding": {
        "findingId": "ef4826b8-e55a-44b9-b460-5c485355371b",
        "taskId": "873ed69c74a9ec8fa9b8e88e9abc4661",
        "checkName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
        "taskStartTime": 1576012045.745,
        "findingTime": 1576012046.168,
        "severity": "CRITICAL",
        "nonCompliantResource": {
            "resourceType": "IOT_POLICY",
            "resourceIdentifier": {
                "policyVersionIdentifier": {
                    "policyName": "smp-ggrass-group_Core-policy",
                    "policyVersionId": "1"
                }
            }
         },
        "reasonForNonCompliance": "Policy allows broad access to IoT data plane
 actions: [iot:Subscribe, iot:Connect, iot:GetThingShadow, iot:DeleteThingShadow,
 iot:UpdateThingShadow, iot:Publish].",
        "reasonForNonComplianceCode":
 "ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS"
    }
}
```

Pour plus d'informations, consultez la section Vérifier les résultats d'audit (commandes d'audit) dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeAuditFindingla section Référence des AWS CLI commandes.

`describe-audit-mitigation-actions-task`

L'exemple de code suivant montre comment utiliser`describe-audit-mitigation-actions-task`.

AWS CLI

Pour afficher les détails d'une tâche d'audit relative aux mesures d'atténuation

L'`describe-audit-mitigation-actions-task`exemple suivant montre les détails de la tâche spécifiée, à laquelle le `ResetPolicyVersionAction` a été appliqué à un résultat. Les résultats incluent les dates de début et de fin de la tâche, le nombre de résultats ciblés (et le résultat) et la définition de l'action appliquée dans le cadre de cette tâche.

```
aws iot describe-audit-mitigation-actions-task \
    --task-id ResetPolicyTask01
```

Sortie :

```
{
    "taskStatus": "COMPLETED",
    "startTime": "2019-12-10T15:13:19.457000-08:00",
    "endTime": "2019-12-10T15:13:19.947000-08:00",
    "taskStatistics": {
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "totalFindingsCount": 1,
            "failedFindingsCount": 0,
            "succeededFindingsCount": 1,
            "skippedFindingsCount": 0,
            "canceledFindingsCount": 0
        }
    },
    "target": {
        "findingIds": [
            "ef4826b8-e55a-44b9-b460-5c485355371b"
        ]
    },
    "auditCheckToActionsMapping": {
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": [
```

```
            "ResetPolicyVersionAction"
        ]
    },
    "actionsDefinition": [
        {
            "name": "ResetPolicyVersionAction",
            "id": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
            "roleArn": "arn:aws:iam::123456789012:role/service-role/
ReplacePolicyVersionRole",
            "actionParams": {
                "replaceDefaultPolicyVersionParams": {
                    "templateName": "BLANK_POLICY"
                }
            }
        }
    ]
}
```

Pour plus d'informations, voir DescribeAuditMitigationActionsTask (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeAuditMitigationActionsTaskla section Référence des AWS CLI commandes.

`describe-audit-suppression`

L'exemple de code suivant montre comment utiliser`describe-audit-suppression`.

AWS CLI

Pour obtenir des informations sur la suppression d'un résultat d'audit

L'`describe-audit-suppression`exemple suivant répertorie les détails relatifs à la suppression d'un résultat d'audit.

```
aws iot describe-audit-task \
    --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

Sortie :

```
{
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK",
```

```
    "taskStartTime": 1596168096.157,
    "taskStatistics": {
        "totalChecks": 1,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 0,
        "nonCompliantChecks": 1,
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "auditDetails": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_NON_COMPLIANT",
            "checkCompliant": false,
            "totalResourcesCount": 195,
            "nonCompliantResourcesCount": 2
        }
    }
}
```

Pour plus d'informations, consultez la section Audit sur les suppressions dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeAuditSuppressionla section Référence des AWS CLI commandes.

`describe-audit-task`

L'exemple de code suivant montre comment utiliser`describe-audit-task`.

AWS CLI

Pour obtenir des informations sur une instance d'audit

L'`describe-audit-task`exemple suivant permet d'obtenir des informations sur une instance d'un audit AWS IoT Device Defender. Si l'audit est terminé, les statistiques récapitulatives de l'exécution sont incluses dans les résultats.

```
aws iot describe-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

Sortie :

```
{
    "taskStatus": "COMPLETED",
    "taskType": "ON_DEMAND_AUDIT_TASK",
    "taskStartTime": 1560356923.434,
    "taskStatistics": {
        "totalChecks": 3,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 3,
        "nonCompliantChecks": 0,
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "auditDetails": {
        "CA_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_COMPLIANT",
            "checkCompliant": true,
            "totalResourcesCount": 0,
            "nonCompliantResourcesCount": 0
        },
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_COMPLIANT",
            "checkCompliant": true,
            "totalResourcesCount": 6,
            "nonCompliantResourcesCount": 0
        },
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "checkRunStatus": "COMPLETED_COMPLIANT",
            "checkCompliant": true,
            "totalResourcesCount": 0,
            "nonCompliantResourcesCount": 0
        }
    }
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeAuditTaskla section Référence des AWS CLI commandes.

`describe-authorizer`

L'exemple de code suivant montre comment utiliser`describe-authorizer`.

AWS CLI

Pour obtenir des informations sur un autorisateur personnalisé

L'`describe-authorizer`exemple suivant affiche les détails de l'autorisateur personnalisé spécifié.

```
aws iot describe-authorizer \
    --authorizer-name CustomAuthorizer
```

Sortie :

```
{
    "authorizerDescription": {
        "authorizerName": "CustomAuthorizer",
        "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer",
        "authorizerFunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction",
        "tokenKeyName": "MyAuthToken",
        "tokenSigningPublicKeys": {
            "FIRST_KEY": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb5O3kRSmyIwTzwzm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAgTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n
+GqsNvAQ9Jy1wI2VrEa1OfL8flDB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx\n50g2LW2Oy9R/
dmqtDmJiVP97Z4GykxPvwlYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
 KEY-----"
        },
        "status": "ACTIVE",
        "creationDate": 1571245658.069,
        "lastModifiedDate": 1571245658.069
    }
}
```

Pour plus d'informations, consultez DescribeAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir DescribeAuthorizerla section Référence des AWS CLI commandes.

`describe-billing-group`

L'exemple de code suivant montre comment utiliser`describe-billing-group`.

AWS CLI

Pour obtenir des informations sur un groupe de facturation

L'`describe-billing-group`exemple suivant permet d'obtenir des informations pour le groupe de facturation spécifié.

```
aws iot describe-billing-group --billing-group-name GroupOne
```

Sortie :

```
{
    "billingGroupName": "GroupOne",
    "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562",
    "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
    "version": 1,
    "billingGroupProperties": {},
    "billingGroupMetadata": {
        "creationDate": 1560199355.378
    }
}
```

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir DescribeBillingGroupla section Référence des AWS CLI commandes.

`describe-ca-certificate`

L'exemple de code suivant montre comment utiliser`describe-ca-certificate`.

AWS CLI

Pour obtenir des informations sur un certificat CA

L'`describe-ca-certificate`exemple suivant affiche les détails du certificat CA spécifié.

```
aws iot describe-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467
```

Sortie :

```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
        "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
        "status": "INACTIVE",
        "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPWXl8taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTAlVTMQ8wDQYDVQQKDAZBbWF6b24wHhcNMTkwOTI0MjEzMTE1WhcNMjkwOTIx
\nMjEzMTE1WjAeMQswCQYDVQQGEwJVUzEPMA0GA1UECgwGQW1hem9uMIIBIjANBgkq
\nhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzd3R3ioalCS0MhFWfBrVGR036EK07UAf
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18VzlFtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMawOjhF4LsculKG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWWvUskzxnbO1F2WCshbEJvqY8fIWtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJwOQxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+nO4jD5cqaoEPtzoeFUEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJslxwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3IJgOnd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcBlCFwH1yakwjJL1dR4yarnrfxgI
\nEqP4AOYVimAVoQ5FBwnloHe16+3qtDiblU9DeXBUCtS55EcfrEXAMPLEYtXdqU5C
\nU9ia4KAjV0dxW1+EFYMwX5eGeb0gDTNHBylV6B/fOSZiQAwDYp4x3B+gAP+a/bWB
\nu1umOqtBdWe6L6/83L+JhaTByqV25iVJ4c/UZUnG8926wUlDM9zQvEXuEVvzZ7+m\n4PSNqst/
nVOvnLpoG4e0WgcJgANuB33CSWtjWSuYsbhmqQRknGhREXAMPLEZT4fm\nfo0e\n-----END
 CERTIFICATE-----\n",
        "ownedBy": "123456789012",
        "creationDate": 1569365372.053,
        "autoRegistrationStatus": "DISABLE",
        "lastModifiedDate": 1569365372.053,
        "customerVersion": 1,
        "generationId": "c5c2eb95-140b-4f49-9393-6aaac85b2a90",
        "validity": {
            "notBefore": 1569360675.0,
            "notAfter": 1884720675.0
        }
    }
```

```
}
```

Pour plus d'informations, consultez la section Décrire CACertificate dans le manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir DescribeCaCertificatela section Référence des AWS CLI commandes.

`describe-certificate`

L'exemple de code suivant montre comment utiliser`describe-certificate`.

AWS CLI

Pour obtenir des informations sur un certificat

L'`describe-certificate`exemple suivant affiche les détails du certificat spécifié.

```
aws iot describe-certificate \
    --certificate-
id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

Sortie :

```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
        "certificateId":
 "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
        "status": "ACTIVE",
        "certificatePem": "-----BEGIN CERTIFICATE-----
MIICiTEXAMPLEQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDEXAMPLElMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNEXAMPLEdBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBEXAMPLEz
b2xEXAMPLEYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8EXAMPLEZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYEXAMPLEpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7EXAMPLEGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4
```

```
nUhVVxYUnEXAMPLE8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
        "ownedBy": "123456789012",
        "creationDate": 1541022751.983,
        "lastModifiedDate": 1541022751.983,
        "customerVersion": 1,
        "transferData": {},
        "generationId": "6974fbed-2e61-4114-bc5e-4204cc79b045",
        "validity": {
            "notBefore": 1541022631.0,
            "notAfter": 2524607999.0
        }
    }
}
```

Pour plus d'informations, consultez DescribeCertificatele manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir DescribeCertificatela section Référence des AWS CLI commandes.

`describe-custom-metric`

L'exemple de code suivant montre comment utiliser`describe-custom-metric`.

AWS CLI

Pour obtenir des informations sur une métrique personnalisée de Device Defender

L'`describe-custom-metric`exemple suivant permet d'obtenir des informations sur une métrique personnalisée nomméemy`CustomMetric`.

```
aws iot describe-custom-metric \
    --metric-name myCustomMetric
```

Sortie :

```
{
    "metricName": "myCustomMetric",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/myCustomMetric",
    "metricType": "number",
```

```
    "displayName": "My custom metric",
    "creationDate": 2020-11-17T23:02:12.879000-09:00,
    "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00
}
```

Pour plus d'informations, consultez la section Mesures personnalisées dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, voir DescribeCustomMetricla section Référence des AWS CLI commandes.

`describe-default-authorizer`

L'exemple de code suivant montre comment utiliser`describe-default-authorizer`.

AWS CLI

Pour obtenir des informations sur l'autorisateur personnalisé par défaut

L'`describe-default-authorizer`exemple suivant affiche les détails de l'autorisateur personnalisé par défaut.

```
aws iot describe-default-authorizer
```

Sortie :

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

Pour plus d'informations, consultez DescribeDefaultAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir DescribeDefaultAuthorizerla section Référence des AWS CLI commandes.

`describe-dimension`

L'exemple de code suivant montre comment utiliser`describe-dimension`.

AWS CLI

Pour obtenir des informations sur une dimension

L'`describe-dimension`exemple suivant permet d'obtenir des informations sur une dimension nomméeTopicFilterForAuthMessages.

```
aws iot describe-dimension \
    --name TopicFilterForAuthMessages
```

Sortie :

```
{
    "name": "TopicFilterForAuthMessages",
    "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
    "type": "TOPIC_FILTER",
    "stringValues": [
        "device/+/auth"
    ],
    "creationDate": 1578620223.255,
    "lastModifiedDate": 1578620223.255
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeDimensionla section Référence des AWS CLI commandes.

`describe-domain-configuration`

L'exemple de code suivant montre comment utiliser`describe-domain-configuration`.

AWS CLI

Pour décrire la configuration d'un domaine

L'`describe-domain-configuration`exemple suivant affiche les détails de la configuration du domaine spécifié.

```
aws iot describe-domain-configuration \
```

```
    --domain-configuration-name "additionalDataDomain"
```

Sortie :

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
east-1:758EXAMPLE143:domainconfiguration/additionalDataDomain/norpw",
    "domainName": "d055exampleed74y71zfd-ats.beta.us-east-1.iot.amazonaws.com",
    "serverCertificates": [],
    "domainConfigurationStatus": "ENABLED",
    "serviceType": "DATA",
    "domainType": "AWS_MANAGED",
    "lastStatusChangeDate": 1601923783.774
}
```

Pour plus d'informations, consultez la section Points de terminaison configurables dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeDomainConfigurationla section Référence des AWS CLI commandes.

`describe-endpoint`

L'exemple de code suivant montre comment utiliser`describe-endpoint`.

AWS CLI

Exemple 1 : pour obtenir votre point de AWS terminaison actuel

L'`describe-endpoint`exemple suivant récupère le point de AWS terminaison par défaut auquel toutes les commandes sont appliquées.

```
aws iot describe-endpoint
```

Sortie :

```
{
    "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

Pour plus d'informations, consultez DescribeEndpointle Guide du développeur de AWS l'IoT.

Exemple 2 : pour obtenir votre point de terminaison ATS

L'`describe-endpoint`exemple suivant récupère le point de terminaison Amazon Trust Services (ATS).

```
aws iot describe-endpoint \
    --endpoint-type iot:Data-ATS
```

Sortie :

```
{
    "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"
}
```

Pour plus d'informations, consultez la section Certificats X.509 et AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeEndpointla section Référence des AWS CLI commandes.

`describe-event-configurations`

L'exemple de code suivant montre comment utiliser`describe-event-configurations`.

AWS CLI

Pour afficher les types d'événements publiés

L'`describe-event-configurations`exemple suivant répertorie la configuration qui contrôle les événements générés lors de l'ajout, de la mise à jour ou de la suppression d'un élément.

```
aws iot describe-event-configurations
```

Sortie :

```
{
    "eventConfigurations": {
        "CA_CERTIFICATE": {
            "Enabled": false
```

```
        },
        "CERTIFICATE": {
            "Enabled": false
        },
        "JOB": {
            "Enabled": false
        },
        "JOB_EXECUTION": {
            "Enabled": false
        },
        "POLICY": {
            "Enabled": false
        },
        "THING": {
            "Enabled": false
        },
        "THING_GROUP": {
            "Enabled": false
        },
        "THING_GROUP_HIERARCHY": {
            "Enabled": false
        },
        "THING_GROUP_MEMBERSHIP": {
            "Enabled": false
        },
        "THING_TYPE": {
            "Enabled": false
        },
        "THING_TYPE_ASSOCIATION": {
            "Enabled": false
        }
    }
}
```

Pour plus d'informations, consultez la section Messages d'événements dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeEventConfigurationsla section Référence des AWS CLI commandes.

`describe-index`

L'exemple de code suivant montre comment utiliser`describe-index`.

AWS CLI

Pour récupérer l'état actuel de l'index des objets

L'`describe-index`exemple suivant récupère le statut actuel de l'index des objets.

```
aws iot describe-index \
    --index-name "AWS_Things"
```

Sortie :

```
{
    "indexName": "AWS_Things",
    "indexStatus": "ACTIVE",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

Pour plus d'informations, consultez la section Managing Thing Indexing dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeIndexla section Référence des AWS CLI commandes.

`describe-job-execution`

L'exemple de code suivant montre comment utiliser`describe-job-execution`.

AWS CLI

Pour obtenir les détails d'exécution d'une tâche sur un appareil

L'`describe-job-execution`exemple suivant permet d'obtenir les détails d'exécution de la tâche spécifiée.

```
aws iot describe-job-execution \
    --job-id "example-job-01" \
    --thing-name "MyRaspberryPi"
```

Sortie :

```
{
    "execution": {
```

```
        "jobId": "example-job-01",
        "status": "QUEUED",
        "statusDetails": {},
        "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi",
        "queuedAt": 1560787023.636,
        "lastUpdatedAt": 1560787023.636,
        "executionNumber": 1,
        "versionNumber": 1
    }
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeJobExecutionla section Référence des AWS CLI commandes.

`describe-job`

L'exemple de code suivant montre comment utiliser`describe-job`.

AWS CLI

Pour obtenir le statut détaillé d'une tâche

L'`describe-job`exemple suivant permet d'obtenir le statut détaillé de la tâche dont l'ID est`example-job-01`.

```
aws iot describe-job \
    --job-id "example-job-01"
```

Sortie :

```
{
    "job": {
        "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
        "jobId": "example-job-01",
        "targetSelection": "SNAPSHOT",
        "status": "IN_PROGRESS",
        "targets": [
            "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"
        ],
```

```
        "description": "example job test",
        "presignedUrlConfig": {},
        "jobExecutionsRolloutConfig": {},
        "createdAt": 1560787022.733,
        "lastUpdatedAt": 1560787026.294,
        "jobProcessDetails": {
            "numberOfCanceledThings": 0,
            "numberOfSucceededThings": 0,
            "numberOfFailedThings": 0,
            "numberOfRejectedThings": 0,
            "numberOfQueuedThings": 1,
            "numberOfInProgressThings": 0,
            "numberOfRemovedThings": 0,
            "numberOfTimedOutThings": 0
        },
        "timeoutConfig": {}
    }
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeJobla section Référence des AWS CLI commandes.

`describe-mitigation-action`

L'exemple de code suivant montre comment utiliser`describe-mitigation-action`.

AWS CLI

Pour consulter les détails d'une action d'atténuation définie

L'`describe-mitigation-action`exemple suivant affiche les détails de l'action d'atténuation spécifiée.

```
aws iot describe-mitigation-action \
    --action-name AddThingsToQuarantineGroupAction
```

Sortie :

```
{
    "actionName": "AddThingsToQuarantineGroupAction",
    "actionType": "ADD_THINGS_TO_THING_GROUP",
```

```
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
    "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole",
    "actionParams": {
        "addThingsToThingGroupParams": {
            "thingGroupNames": [
                "QuarantineGroup1"
            ],
            "overrideDynamicGroups": true
        }
    },
    "creationDate": "2019-12-10T11:09:35.999000-08:00",
    "lastModifiedDate": "2019-12-10T11:09:35.999000-08:00"
}
```

Pour plus d'informations, voir DescribeMitigationAction (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeMitigationActionla section Référence des AWS CLI commandes.

`describe-provisioning-template-version`

L'exemple de code suivant montre comment utiliser`describe-provisioning-template-version`.

AWS CLI

Pour décrire la version d'un modèle de provisionnement

L'`describe-provisioning-template-version`exemple suivant décrit une version de modèle de provisionnement.

```
aws iot describe-provisioning-template-version \
    --template-name MyTestProvisioningTemplate \
    --version-id 1
```

Sortie :

```
{
```

```
    "versionId": 1,
    "creationDate": 1589308310.574,
    "templateBody": "{
        \"Parameters\":{
            \"SerialNumber\":{
                \"Type\":\"String\"
            },
            \"AWS::IoT::Certificate::Id\":{
                \"Type\":\"String\"
            }
        },
        \"Resources\":{
            \"certificate\":{
                \"Properties\":{
                    \"CertificateId\":{
                        \"Ref\":\"AWS::IoT::Certificate::Id\"
                    },
                    \"Status\":\"Active\"
                },
                \"Type\":\"AWS::IoT::Certificate\"
            },
            \"policy\":{
                \"Properties\":{
                    \"PolicyName\":\"MyIotPolicy\"
                },
                \"Type\":\"AWS::IoT::Policy\"
            },
            \"thing\":{
                \"OverrideSettings\":{
                    \"AttributePayload\":\"MERGE\",
                    \"ThingGroups\":\"DO_NOTHING\",
                    \"ThingTypeName\":\"REPLACE\"
                },
                \"Properties\":{
                    \"AttributePayload\":{},
                    \"ThingGroups\":[],
                    \"ThingName\":{
                        \"Fn::Join\":[
                            \"\",
                            [
                                \"DemoGroup_\",
                                {\"Ref\":\"SerialNumber\"}
                            ]
                        ]
```

```
                    },
                    \"ThingTypeName\":\"VirtualThings\"
                },
                \"Type\":\"AWS::IoT::Thing\"
            }
        }
    }",
    "isDefaultVersion": true
}
```

Pour plus d'informations, consultez la section Provisionnement d'appareils ne disposant pas de certificats d'appareils à l'aide du provisionnement de flotte dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, voir DescribeProvisioningTemplateVersionla section Référence des AWS CLI commandes.

`describe-provisioning-template`

L'exemple de code suivant montre comment utiliser`describe-provisioning-template`.

AWS CLI

Pour décrire un modèle de provisionnement

L'`describe-provisioning-template`exemple suivant décrit un modèle de provisionnement.

```
aws iot describe-provisioning-template \
    --template-name MyTestProvisioningTemplate
```

Sortie :

```
{
    "templateArn": "arn:aws:iot:us-west-2:57EXAMPLE833:provisioningtemplate/
MyTestProvisioningTemplate",
    "templateName": "MyTestProvisioningTemplate",
    "creationDate": 1589308310.574,
    "lastModifiedDate": 1589308345.539,
    "defaultVersionId": 1,
    "templateBody": "{
        \"Parameters\":{
            \"SerialNumber\":{
                \"Type\":\"String\"
```

```
            },
            \"AWS::IoT::Certificate::Id\":{
                \"Type\":\"String\"
            }
        },
        \"Resources\":{
            \"certificate\":{
                \"Properties\":{
                    \"CertificateId\":{
                        \"Ref\":\"AWS::IoT::Certificate::Id\"
                    },
                    \"Status\":\"Active\"
                },
                \"Type\":\"AWS::IoT::Certificate\"
            },
            \"policy\":{
                \"Properties\":{
                    \"PolicyName\":\"MyIotPolicy\"
                },
                \"Type\":\"AWS::IoT::Policy\"
            },
            \"thing\":{
                \"OverrideSettings\":{
                    \"AttributePayload\":\"MERGE\",
                    \"ThingGroups\":\"DO_NOTHING\",
                    \"ThingTypeName\":\"REPLACE\"
                },
                \"Properties\":{
                    \"AttributePayload\":{},
                    \"ThingGroups\":[],
                    \"ThingName\":{
                        \"Fn::Join\":[
                            \"\",
                            [
                                \"DemoGroup_\",
                                {\"Ref\":\"SerialNumber\"}
                            ]
                        ]
                    },
                    \"ThingTypeName\":\"VirtualThings\"
                },
                \"Type\":\"AWS::IoT::Thing\"
            }
        }
```

```
    }",
    "enabled": true,
    "provisioningRoleArn": "arn:aws:iam::571032923833:role/service-role/IoT_access"
}
```

Pour plus d'informations, consultez la section Provisionnement d'appareils ne disposant pas de certificats d'appareils à l'aide du provisionnement de flotte dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, voir DescribeProvisioningTemplatela section Référence des AWS CLI commandes.

`describe-role-alias`

L'exemple de code suivant montre comment utiliser`describe-role-alias`.

AWS CLI

Pour obtenir des informations sur un alias de rôle AWS IoT

L'`describe-role-alias`exemple suivant affiche les détails de l'alias de rôle spécifié.

```
aws iot describe-role-alias \
    --role-alias LightBulbRole
```

Sortie :

```
{
    "roleAliasDescription": {
        "roleAlias": "LightBulbRole",
        "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/
LightBulbRole",
        "roleArn": "arn:aws:iam::123456789012:role/light_bulb_role_001",
        "owner": "123456789012",
        "credentialDurationSeconds": 3600,
        "creationDate": 1570558643.221,
        "lastModifiedDate": 1570558643.221
    }
}
```

Pour plus d'informations, consultez DescribeRoleAliasle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir DescribeRoleAliasla section Référence des AWS CLI commandes.

`describe-scheduled-audit`

L'exemple de code suivant montre comment utiliser`describe-scheduled-audit`.

AWS CLI

Pour obtenir des informations sur un audit planifié

L'`describe-scheduled-audit`exemple suivant fournit des informations détaillées sur un audit planifié AWS IOT Device Defender nomméAWSIoTDeviceDefenderDailyAudit.

```
aws iot describe-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Sortie :

```
{
    "frequency": "DAILY",
    "targetCheckNames": [
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK",
        "CONFLICTING_CLIENT_IDS_CHECK",
        "DEVICE_CERTIFICATE_SHARED_CHECK",
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK",
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK"
    ],
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSIoTDeviceDefenderDailyAudit"
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeScheduledAuditla section Référence des AWS CLI commandes.

`describe-security-profile`

L'exemple de code suivant montre comment utiliser`describe-security-profile`.

AWS CLI

Pour obtenir des informations sur un profil de sécurité

L'`describe-security-profile`exemple suivant permet d'obtenir des informations sur le profil de sécurité AWS IoT Device Defender nommé `PossibleIssue`.

```
aws iot describe-security-profile \
    --security-profile-name PossibleIssue
```

Sortie :

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
    "securityProfileDescription": "check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128",
    "behaviors": [
        {
            "name": "CellularBandwidth",
            "metric": "aws:message-byte-size",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 128
                },
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        {
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 10
                },
```

```
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        }
    ],
    "version": 1,
    "creationDate": 1560278102.528,
    "lastModifiedDate": 1560278102.528
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeSecurityProfilela section Référence des AWS CLI commandes.

`describe-stream`

L'exemple de code suivant montre comment utiliser`describe-stream`.

AWS CLI

Pour obtenir des informations sur un stream

L'`describe-stream`exemple suivant affiche les détails du flux spécifié.

```
aws iot describe-stream \
    --stream-id stream12345
```

Sortie :

```
{
    "streamInfo": {
        "streamId": "stream12345",
        "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
        "streamVersion": 1,
        "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
        "files": [
            {
                "fileId": "123",
                "s3Location": {
```

```
                    "bucket":"codesign-ota-bucket",
                    "key":"48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
                }
            }
        ],
        "createdAt": 1557863215.995,
        "lastUpdatedAt": 1557863215.995,
        "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"
    }
}
```

Pour plus d'informations, consultez DescribeStreamle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir DescribeStreamla section Référence des AWS CLI commandes.

`describe-thing-group`

L'exemple de code suivant montre comment utiliser`describe-thing-group`.

AWS CLI

Pour obtenir des informations sur un groupe d'objets

L'`describe-thing-group`exemple suivant permet d'obtenir des informations sur le groupe d'objets nommé`HalogenBulbs`.

```
aws iot describe-thing-group \
    --thing-group-name HalogenBulbs
```

Sortie :

```
{
    "thingGroupName": "HalogenBulbs",
    "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",
    "version": 1,
    "thingGroupProperties": {},
    "thingGroupMetadata": {
        "parentGroupName": "LightBulbs",
        "rootToParentThingGroups": [
            {
```

```
                "groupName": "LightBulbs",
                "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
LightBulbs"
            }
        ],
        "creationDate": 1559927609.897
    }
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir DescribeThingGroupla section Référence des AWS CLI commandes.

`describe-thing-type`

L'exemple de code suivant montre comment utiliser`describe-thing-type`.

AWS CLI

Pour obtenir des informations sur un type d'objet

L'`describe-thing-type`exemple suivant affiche des informations sur le type d'objet spécifié défini dans votre AWS compte.

```
aws iot describe-thing-type \
    --thing-type-name "LightBulb"
```

Sortie :

```
{
    "thingTypeName": "LightBulb",
    "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeProperties": {
        "thingTypeDescription": "light bulb type",
        "searchableAttributes": [
            "model",
            "wattage"
        ]
    },
```

```
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1559772562.498
    }
}
```

Pour plus d'informations, consultez la section Types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DescribeThingTypela section Référence des AWS CLI commandes.

`describe-thing`

L'exemple de code suivant montre comment utiliser`describe-thing`.

AWS CLI

Pour afficher des informations détaillées sur un objet

L'`describe-thing`exemple suivant affiche des informations sur un objet (appareil) défini dans le registre AWS IoT de votre AWS compte.

est-ce que je ne décrivais pas --thing-name « » MyLightBulb

Sortie :

```
{
    "defaultClientId": "MyLightBulb",
    "thingName": "MyLightBulb",
    "thingId": "40da2e73-c6af-406e-b415-15acae538797",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "thingTypeName": "LightBulb",
    "attributes": {
        "model": "123",
        "wattage": "75"
    },
    "version": 1
}
```

Pour plus d'informations, consultez la section Comment gérer les objets avec le registre dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir DescribeThingla section Référence des AWS CLI commandes.

`detach-policy`

L'exemple de code suivant montre comment utiliser`detach-policy`.

AWS CLI

Exemple 1 : pour détacher une politique AWS IoT d'un groupe d'objets

L'`detach-policy`exemple suivant détache la politique spécifiée d'un groupe d'objets et, par extension, de tous les éléments de ce groupe et de tous les groupes enfants du groupe.

```
aws iot detach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "MyFirstGroup_Core-policy"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

Exemple 2 : pour détacher une politique AWS IoT d'un certificat d'appareil

L'`detach-policy`exemple suivant détache la TemperatureSensorPolicy politique d'un certificat d'appareil identifié par l'ARN.

```
aws iot detach-policy \
    --policy-name TemperatureSensorPolicy \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, voir DetachPolicyla section Référence des AWS CLI commandes.

`detach-security-profile`

L'exemple de code suivant montre comment utiliser`detach-security-profile`.

AWS CLI

Pour dissocier un profil de sécurité d'une cible

L'`detach-security-profile`exemple suivant supprime l'association entre le profil de sécurité AWS IoT Device Defender nommé `Testprofile` et la cible de tous les objets enregistrés.

```
aws iot detach-security-profile \
    --security-profile-name Testprofile \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
registered-things"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DetachSecurityProfilela section Référence des AWS CLI commandes.

`detach-thing-principal`

L'exemple de code suivant montre comment utiliser`detach-thing-principal`.

AWS CLI

Pour détacher une chose certificate/principal d'une chose

L'`detach-thing-principal`exemple suivant supprime un certificat représentant un mandant de l'objet spécifié.

```
aws iot detach-thing-principal \
    --thing-name "MyLightBulb" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Comment gérer les objets avec le registre dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir DetachThingPrincipalla section Référence des AWS CLI commandes.

`disable-topic-rule`

L'exemple de code suivant montre comment utiliser`disable-topic-rule`.

AWS CLI

Pour désactiver une règle de sujet

L'`disable-topic-rule`exemple suivant désactive la règle de rubrique spécifiée.

```
aws iot disable-topic-rule \
    --rule-name "MyPlantPiMoistureAlertRule"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Visualisation de vos règles dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir DisableTopicRulela section Référence des AWS CLI commandes.

`enable-topic-rule`

L'exemple de code suivant montre comment utiliser`enable-topic-rule`.

AWS CLI

Pour activer une règle de sujet

L'`enable-topic-rule`exemple suivant active (ou réactive) la règle de rubrique spécifiée.

```
aws iot enable-topic-rule \
    --rule-name "MyPlantPiMoistureAlertRule"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Visualisation de vos règles dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir EnableTopicRulela section Référence des AWS CLI commandes.

`get-behavior-model-training-summaries`

L'exemple de code suivant montre comment utiliser`get-behavior-model-training-summaries`.

AWS CLI

Pour répertorier l'état du modèle de formation ML Detect Security Profile d'un Device Defender

L'`get-behavior-model-training-summaries`exemple suivant répertorie l'état d'entraînement du modèle pour les comportements configurés dans le profil de sécurité choisi. Pour chaque comportement, le nom, le statut du modèle et le pourcentage de points de données collectés sont répertoriés.

```
aws iot get-behavior-model-training-summaries \
    --security-profile-name MySecuirtyProfileName
```

Sortie :

```
{
    "summaries": [
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Messages_sent_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Messages_received_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Authorization_failures_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
```

```
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Message_size_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Connection_attempts_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySPNoALerts",
            "behaviorName": "Disconnects_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        }
    ]
}
```

Pour plus d'informations, voir GetBehaviorModelTrainingSummaries (Detect Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetBehaviorModelTrainingSummariesla section Référence des AWS CLI commandes.

`get-cardinality`

L'exemple de code suivant montre comment utiliser`get-cardinality`.

AWS CLI

Pour renvoyer le nombre approximatif de valeurs uniques correspondant à la requête

Vous pouvez utiliser le script de configuration suivant pour créer 10 éléments représentant 10 capteurs de température. Chaque nouvel objet possède 3 attributs.

```
# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
Racks=(Rack1 Rack1 Rack2 Rack2 Rack3 Rack4 Rack5 Rack6 Rack6 Rack6)
IsNormal=(true true true true true true false false false false)
```

```
for ((i=0; i<10 ; i++))
do
  thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
 attributes="{temperature=${Temperatures[i]},rackId=${Racks[i]},stateNormal=
${IsNormal[i]}}")
  aws iot describe-thing --thing-name "TempSensor$i"
done
```

Exemple de sortie du script de configuration :

```
{
    "version": 1,
    "thingName": "TempSensor0",
    "defaultClientId": "TempSensor0",
    "attributes": {
        "rackId": "Rack1",
        "stateNormal": "true",
        "temperature": "70"
    },
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
    "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

L'`get-cardinality`exemple suivant interroge les 10 capteurs créés par le script de configuration et renvoie le nombre de racks équipés de capteurs de température signalant des valeurs de température anormales. Si la valeur de température est inférieure à 60 ou supérieure à 80, le capteur de température est dans un état anormal.

```
aws iot get-cardinality \
    --aggregation-field "attributes.rackId" \
    --query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

Sortie :

```
{
    "cardinality": 2
}
```

Pour plus d'informations, consultez la section Requête de données agrégées < https:// docs.aws.amazon.com/iot/ latest/developerguide/index -aggregate.html> dans le guide du développeur de l'IoT.AWS

- Pour plus de détails sur l'API, voir GetCardinalityla section Référence des AWS CLI commandes.

`get-effective-policies`

L'exemple de code suivant montre comment utiliser`get-effective-policies`.

AWS CLI

Pour répertorier les politiques qui affectent un objet

L'`get-effective-policies`exemple suivant répertorie les politiques qui affectent l'objet spécifié, y compris les politiques associées aux groupes auxquels il appartient.

```
aws iot get-effective-policies \
    --thing-name TemperatureSensor-001 \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Sortie :

```
{
    "effectivePolicies": [
        {
            "policyName": "TemperatureSensorPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
            "policyDocument": "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Publish\",
                            \"iot:Receive\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
                            \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
                        ]
                    },
                    {
```

```
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Subscribe\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_1\",
                            \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_2\"
                        ]
                    },
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Connect\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub
\"
                        ]
                    }
                ]
            }"
        }
    ]
}
```

Pour plus d'informations, consultez la section Get Effective Policies for a Thing du Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetEffectivePolicieslа section Référence des AWS CLI commandes.

`get-indexing-configuration`

L'exemple de code suivant montre comment utiliser`get-indexing-configuration`.

AWS CLI

Pour obtenir la configuration d'indexation de l'objet

L'`get-indexing-configuration`exemple suivant permet d'obtenir les données de configuration actuelles pour AWS l'indexation des flottes IoT.

```
aws iot get-indexing-configuration
```

Sortie :

```
{
    "thingIndexingConfiguration": {
        "thingIndexingMode": "OFF",
        "thingConnectivityIndexingMode": "OFF"
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    }
}
```

Pour plus d'informations, consultez la section Managing Thing Indexing dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetIndexingConfigurationla section Référence des AWS CLI commandes.

`get-job-document`

L'exemple de code suivant montre comment utiliser`get-job-document`.

AWS CLI

Pour récupérer le document correspondant à une tâche

L'`get-job-document`exemple suivant affiche les détails du document correspondant à la tâche dont l'ID est`example-job-01`.

```
aws iot get-job-document \
    --job-id "example-job-01"
```

Sortie :

```
{
    "document": "\n{\n    \"operation\":\"customJob\",\n    \"otherInfo\":
\"someValue\"\n}\n"
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetJobDocumentla section Référence des AWS CLI commandes.

`get-logging-options`

L'exemple de code suivant montre comment utiliser`get-logging-options`.

AWS CLI

Pour obtenir les options de journalisation

L'`get-logging-options`exemple suivant montre les options de journalisation actuelles de votre AWS compte.

```
aws iot get-logging-options
```

Sortie :

```
{
    "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",
    "logLevel": "ERROR"
}
```

Pour plus d'informations, voir le titre dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetLoggingOptionsla section Référence des AWS CLI commandes.

`get-ota-update`

L'exemple de code suivant montre comment utiliser`get-ota-update`.

AWS CLI

Pour récupérer des informations sur une mise à jour OTA

L'`get-ota-update`exemple suivant affiche les détails de la mise à jour OTA spécifiée.

```
aws iot get-ota-update \
    --ota-update-id ota12345
```

Sortie :


```
{
    "otaUpdateInfo": {
        "otaUpdateId": "ota12345",
        "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
        "creationDate": 1557863215.995,
        "lastModifiedDate": 1557863215.995,
        "description": "A critical update needed right away.",
        "targets": [
           "device1",
           "device2",
           "device3",
           "device4"
        ],
        "targetSelection": "SNAPSHOT",
        "protocols": ["HTTP"],
        "awsJobExecutionsRolloutConfig": {
           "maximumPerMinute": 10
        },
        "otaUpdateFiles": [
            {
                "fileName": "firmware.bin",
                "fileLocation": {
                    "stream": {
                        "streamId": "004",
                        "fileId":123
                    }
                },
                "codeSigning": {
                    "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
                }
            }
        ],
        "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
        "otaUpdateStatus": "CREATE_COMPLETE",
        "awsIotJobId": "job54321",
        "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/job54321",
        "errorInfo": {
        }
```

```
    }
}
```

Pour plus d'informations, consultez la section Guide de référence de l'API Get OTAUpdate in the AWS IoT.

- Pour plus de détails sur l'API, voir GetOtaUpdatela section Référence des AWS CLI commandes.

`get-percentiles`

L'exemple de code suivant montre comment utiliser`get-percentiles`.

AWS CLI

Pour regrouper les valeurs agrégées correspondant à la requête en groupes percentiles

Vous pouvez utiliser le script de configuration suivant pour créer 10 éléments représentant 10 capteurs de température. Chaque nouvel objet possède un attribut.

```
# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
for ((i=0; i<10 ; i++))
do
    thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
 attributes="{temperature=${Temperatures[i]}}")
    aws iot describe-thing --thing-name "TempSensor$i"
done
```

Exemple de sortie du script de configuration :

```
{
    "version": 1,
    "thingName": "TempSensor0",
    "defaultClientId": "TempSensor0",
    "attributes": {
        "temperature": "70"
    },
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
    "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

L'`get-percentiles`exemple suivant interroge les 10 capteurs créés par le script de configuration et renvoie une valeur pour chaque groupe de percentiles spécifié. Le groupe de percentiles « 10 » contient la valeur de champ agrégée qui apparaît dans environ 10 % des valeurs correspondant à la requête. Dans le résultat suivant, {"pourcentage » : 10,0, « valeur » : 67,7} signifie qu'environ 10,0 % des valeurs de température sont inférieures à 67,7.

```
aws iot get-percentiles \
    --aggregation-field "attributes.temperature" \
    --query-string "thingName:TempSensor*" \
    --percents 10 25 50 75 90
```

Sortie :

```
{
    "percentiles": [
        {
            "percent": 10.0,
            "value": 67.7
        },
        {
            "percent": 25.0,
            "value": 71.25
        },
        {
            "percent": 50.0,
            "value": 73.5
        },
        {
            "percent": 75.0,
            "value": 91.5
        },
        {
            "percent": 90.0,
            "value": 98.1
        }
    ]
}
```

Pour plus d'informations, consultez la section Requête de données agrégées dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetPercentilesla section Référence des AWS CLI commandes.

`get-policy-version`

L'exemple de code suivant montre comment utiliser`get-policy-version`.

AWS CLI

Pour obtenir des informations sur une version spécifique d'une politique

L'`get-policy-version`exemple suivant permet d'obtenir des informations sur la première version de la politique spécifiée.

```
aws iot get-policy \
    --policy-name UpdateDeviceCertPolicy
    --policy-version-id "1"
```

Sortie :

```
{
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
    "policyName": "UpdateDeviceCertPolicy",
    "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
 \"Allow\", \"Action\":  \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
    "policyVersionId": "1",
    "isDefaultVersion": false,
    "creationDate": 1559925941.924,
    "lastModifiedDate": 1559926175.458,
    "generationId":
 "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"
}
```

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetPolicyVersionla section Référence des AWS CLI commandes.

`get-policy`

L'exemple de code suivant montre comment utiliser`get-policy`.

AWS CLI

Pour obtenir des informations sur la version par défaut d'une politique

L'`get-policy`exemple suivant permet de récupérer des informations sur la version par défaut de la politique spécifiée.

```
aws iot get-policy \
    --policy-name UpdateDeviceCertPolicy
```

Sortie :

```
{
    "policyName": "UpdateDeviceCertPolicy",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
    "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
 \"Allow\", \"Action\":  \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
    "defaultVersionId": "2",
    "creationDate": 1559925941.924,
    "lastModifiedDate": 1559925941.924,
    "generationId":
 "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"
}
```

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetPolicyla section Référence des AWS CLI commandes.

`get-registration-code`

L'exemple de code suivant montre comment utiliser`get-registration-code`.

AWS CLI

Pour obtenir le code d'enregistrement AWS spécifique à votre compte

L'`get-registration-code`exemple suivant permet de récupérer le code d'enregistrement AWS spécifique à votre compte.

```
aws iot get-registration-code
```

Sortie :

```
{
    "registrationCode":
 "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

Pour plus d'informations, consultez la section Utiliser votre propre certificat dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetRegistrationCodela section Référence des AWS CLI commandes.

`get-statistics`

L'exemple de code suivant montre comment utiliser`get-statistics`.

AWS CLI

Pour rechercher des données agrégées dans l'index des appareils

L'`get-statistics`exemple suivant renvoie le nombre d'objets dotés d'une propriété appelée `connectivity.connected` set to `false` (c'est-à-dire le nombre d'appareils non connectés) dans leur ombre d'appareil.

```
aws iot get-statistics \
    --index-name AWS_Things \
    --query-string "connectivity.connected:false"
```

Sortie :

```
{
    "statistics": {
        "count": 6
    }
}
```

Pour plus d'informations, consultez la section Obtenir des statistiques sur votre parc d'appareils dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetStatisticsla section Référence des AWS CLI commandes.

`get-topic-rule-destination`

L'exemple de code suivant montre comment utiliser`get-topic-rule-destination`.

AWS CLI

Pour obtenir une destination de règle de sujet

L'`get-topic-rule-destination`exemple suivant permet d'obtenir des informations sur la destination d'une règle de sujet.

```
aws iot get-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Sortie :

```
{
    "topicRuleDestination": {
        "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "status": "DISABLED",
        "httpUrlProperties": {
            "confirmationUrl": "https://example.com"
        }
    }
}
```

Pour plus d'informations, consultez la section Travailler avec des destinations de règles thématiques dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetTopicRuleDestinationla section Référence des AWS CLI commandes.

`get-topic-rule`

L'exemple de code suivant montre comment utiliser`get-topic-rule`.

AWS CLI

Pour obtenir des informations sur une règle

L'`get-topic-rule`exemple suivant permet d'obtenir des informations sur la règle spécifiée.

```
aws iot get-topic-rule \
    --rule-name MyRPiLowMoistureAlertRule
```

Sortie :

```
{
    "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/MyRPiLowMoistureAlertRule",
    "rule": {
        "ruleName": "MyRPiLowMoistureAlertRule",
        "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
 state.reported.moisture = 'low'\n                    ",
        "description": "Sends an alert whenever soil moisture level readings are too
 low.",
        "createdAt": 1558624363.0,
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
                    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
                    "messageFormat": "RAW"
                }
            }
        ],
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23"
    }
}
```

Pour plus d'informations, consultez la section Visualisation de vos règles dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir GetTopicRulela section Référence des AWS CLI commandes.

`get-v2-logging-options`

L'exemple de code suivant montre comment utiliser`get-v2-logging-options`.

AWS CLI

Pour répertorier les options de journalisation actuelles

L'`get-v2-logging-options`exemple suivant répertorie les options de journalisation actuelles pour AWS l'IoT.

```
aws iot get-v2-logging-options
```

Sortie :

```
{
    "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",
    "defaultLogLevel": "WARN",
    "disableAllLogs": false
}
```

Pour plus d'informations, voir le titre dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, consultez GetV2 LoggingOptions dans AWS CLI Command Reference.

`list-active-violations`

L'exemple de code suivant montre comment utiliser`list-active-violations`.

AWS CLI

Pour répertorier les violations actives

L'`list-active-violations`exemple suivant répertorie toutes les violations pour le profil de sécurité spécifié.

```
aws iot list-active-violations \
    --security-profile-name Testprofile
```

Sortie :

```
{
    "activeViolations": [
        {
            "violationId": "174db59167fa474c80a652ad1583fd44",
            "thingName": "iotconsole-1560269126751-1",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560293700.0,
            "violationStartTime": 1560279000.0
        },
        {
            "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "CellularBandwidth",
                "metric": "aws:message-byte-size",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 128
                    },
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 110
```

```
            },
            "lastViolationTime": 1560369000.0,
            "violationStartTime": 1560276600.0
        },
        {
            "violationId": "74aa393adea02e6648f3ac362beed55e",
            "thingName": "iotconsole-1560269232412-2",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560276600.0,
            "violationStartTime": 1560276600.0
        },
        {
            "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
```

```
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560369000.0,
            "violationStartTime": 1560276600.0
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListActiveViolationsla section Référence des AWS CLI commandes.

`list-attached-policies`

L'exemple de code suivant montre comment utiliser`list-attached-policies`.

AWS CLI

Exemple 1 : pour répertorier les politiques associées à un groupe

L'`list-attached-policies`exemple suivant répertorie les politiques associées au groupe spécifié.

```
aws iot list-attached-policies \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
```

Sortie :

```
{
    "policies": [
        {
            "policyName": "UpdateDeviceCertPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
UpdateDeviceCertPolicy"
        }
    ]
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

Exemple 2 : pour répertorier les politiques associées à un certificat d'appareil

L'`list-attached-policies`exemple suivant répertorie les politiques AWS IoT associées au certificat de l'appareil. Le certificat est identifié par son ARN.

```
aws iot list-attached-policies \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Sortie :

```
{
    "policies": [
        {
            "policyName": "TemperatureSensorPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy"
        }
    ]
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir ListAttachedPolicies la section Référence des AWS CLI commandes.

`list-audit-findings`

L'exemple de code suivant montre comment utiliser`list-audit-findings`.

AWS CLI

Exemple 1 : Pour répertorier tous les résultats d'un audit

L'`list-audit-findings`exemple suivant répertorie tous les résultats d'un audit AWS IoT Device Defender avec un ID de tâche spécifié.

```
aws iot list-audit-findings \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

Sortie :

```
{
```

```
    "findings": []
}
```

Exemple 2 : Pour répertorier les résultats d'un type de contrôle d'audit

L'`list-audit-findings`exemple suivant montre les résultats des audits AWS IoT Device Defender réalisés entre le 5 juin 2019 et le 19 juin 2019 dans lesquels des appareils partagent un certificat d'appareil. Lorsque vous spécifiez un nom de chèque, vous devez indiquer une heure de début et de fin.

```
aws iot list-audit-findings \
    --check-name DEVICE_CERTIFICATE_SHARED_CHECK \
    --start-time 1559747125 \
    --end-time 1560962028
```

Sortie :

```
{
    "findings": [
        {
            "taskId": "eeef61068b0eb03c456d746c5a26ee04",
            "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
            "taskStartTime": 1560161017.172,
            "findingTime": 1560161017.592,
            "severity": "CRITICAL",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId":
 "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
                }
            },
            "relatedResources": [
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1560086374068"
                    }
                },
```

```
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1560081552187",
                        "DISCONNECTION_TIME": "1560086371552"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559289863631",
                        "DISCONNECTION_TIME": "1560081532716"
                    }
                }
            ],
            "reasonForNonCompliance": "Certificate shared by one or more devices.",
            "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
        },
        {
            "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
            "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
            "taskStartTime": 1559988217.27,
            "findingTime": 1559988217.655,
            "severity": "CRITICAL",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
                }
            },
            "relatedResources": [
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "xShGENLW"
                    },
                    "additionalInfo": {
```

```
                        "CONNECTION_TIME": "1559972350825"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "xShGENLW"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559255062002",
                        "DISCONNECTION_TIME": "1559972350616"
                    }
                }
            ],
            "reasonForNonCompliance": "Certificate shared by one or more devices.",
            "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
        },
        {
            "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
            "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
            "taskStartTime": 1559901817.31,
            "findingTime": 1559901817.767,
            "severity": "CRITICAL",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
                }
            },
            "relatedResources": [
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "TvnQoEoU"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559826729768"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "TvnQoEoU"
```

```
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559345920964",
                        "DISCONNECTION_TIME": "1559826728402"
                    }
                }
            ],
            "reasonForNonCompliance": "Certificate shared by one or more devices.",
            "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
        }
    ]
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir ListAuditFindingsla section Référence des AWS CLI commandes.

`list-audit-mitigation-actions-executions`

L'exemple de code suivant montre comment utiliser`list-audit-mitigation-actions-executions`.

AWS CLI

Pour répertorier les détails de l'exécution d'une action d'atténuation d'un audit

Une tâche d'atténuation d'audit applique une action d'atténuation à un ou plusieurs résultats d'un audit AWS IoT Device Defender. L'`list-audit-mitigation-actions-executions`exemple suivant répertorie les détails de la tâche d'action d'atténuation avec le résultat spécifié `taskId` et pour le résultat spécifié.

```
aws iot list-audit-mitigation-actions-executions \
    --task-id myActionsTaskId \
    --finding-id 0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464
```

Sortie :

```
{
    "actionsExecutions": [
```

```
        {
            "taskId": "myActionsTaskId",
            "findingId": "0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464",
            "actionName": "ResetPolicyVersionAction",
            "actionId": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
            "status": "COMPLETED",
            "startTime": "2019-12-10T15:19:13.279000-08:00",
            "endTime": "2019-12-10T15:19:13.337000-08:00"
        }
    ]
}
```

Pour plus d'informations, voir ListAuditMitigationActionsExecutions (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir ListAuditMitigationActionsExecutionsla section Référence des AWS CLI commandes.

`list-audit-mitigation-actions-tasks`

L'exemple de code suivant montre comment utiliser`list-audit-mitigation-actions-tasks`.

AWS CLI

Pour répertorier les tâches d'audit relatives aux mesures d'atténuation

L'`list-audit-mitigation-actions-tasks`exemple suivant répertorie les mesures d'atténuation qui ont été appliquées aux résultats au cours de la période spécifiée.

```
aws iot list-audit-mitigation-actions-tasks \
    --start-time 1594157400 \
    --end-time 1594157430
```

Sortie :

```
{
    "tasks": [
        {
                "taskId": "0062f2d6-3999-488f-88c7-bef005414103",
                "startTime": "2020-07-07T14:30:15.172000-07:00",
            "taskStatus": "COMPLETED"
        }
```

```
    ]
}
```

Pour plus d'informations, voir ListAuditMitigationActionsTasks (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir ListAuditMitigationActionsTasksla section Référence des AWS CLI commandes.

`list-audit-suppressions`

L'exemple de code suivant montre comment utiliser`list-audit-suppressions`.

AWS CLI

Pour répertorier toutes les suppressions des résultats d'audit

L'`list-audit-suppressions`exemple suivant répertorie toutes les suppressions de résultats d'audit actives.

```
aws iot list-audit-suppressions
```

Sortie :

```
{
    "suppressions": [
        {
        "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "resourceIdentifier": {
                "deviceCertificateId": "c7691e<shortened>"
            },
        "expirationDate": 1597881600.0,
        "suppressIndefinitely": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Audit sur les suppressions dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir ListAuditSuppressionsla section Référence des AWS CLI commandes.

`list-audit-tasks`

L'exemple de code suivant montre comment utiliser`list-audit-tasks`.

AWS CLI

Pour répertorier tous les résultats d'un audit

L'`list-audit-tasks`exemple suivant répertorie les tâches d'audit exécutées entre le 5 juin 2019 et le 12 juin 2019.

```
aws iot list-audit-tasks \
    --start-time 1559747125 \
    --end-time 1560357228
```

Sortie :

```
{
    "tasks": [
        {
            "taskId": "a3aea009955e501a31b764abe1bebd3d",
            "taskStatus": "COMPLETED",
            "taskType": "ON_DEMAND_AUDIT_TASK"
        },
        {
            "taskId": "f76b4b5102b632cd9ae38a279c266da1",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "51d9967d9f9ff4d26529505f6d2c444a",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "eeef61068b0eb03c456d746c5a26ee04",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "041c49557b7c7b04c079a49514b55589",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
```

```
        },
        {
            "taskId": "82c7f2afac1562d18a4560be73998acc",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "ac9086b7222a2f5e2e17bb6fd30b3aeb",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        }
    ]
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir ListAuditTasksla section Référence des AWS CLI commandes.

`list-authorizers`

L'exemple de code suivant montre comment utiliser`list-authorizers`.

AWS CLI

Pour répertorier votre autorisateur personnalisé

L'`list-authorizers`exemple suivant répertorie les autorisateurs personnalisés de votre AWS compte.

```
aws iot list-authorizers
```

Sortie :

```
{
    "authorizers": [
        {
            "authorizerName": "CustomAuthorizer",
            "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
        },
        {
            "authorizerName": "CustomAuthorizer2",
            "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
        },
        {
            "authorizerName": "CustomAuthorizer3",
            "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer3"
        }
    ]
}
```

Pour plus d'informations, consultez ListAuthorizersle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir ListAuthorizersla section Référence des AWS CLI commandes.

`list-billing-groups`

L'exemple de code suivant montre comment utiliser`list-billing-groups`.

AWS CLI

Pour répertorier les groupes de facturation pour votre AWS compte et votre région

L'`list-billing-groups`exemple suivant répertorie tous les groupes de facturation définis pour votre AWS compte et votre AWS région.

```
aws iot list-billing-groups
```

Sortie :

```
{
    "billingGroups": [
        {
            "groupName": "GroupOne",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir ListBillingGroupsla section Référence des AWS CLI commandes.

`list-ca-certificates`

L'exemple de code suivant montre comment utiliser`list-ca-certificates`.

AWS CLI

Pour répertorier les certificats CA enregistrés dans votre AWS compte

L'`list-ca-certificates`exemple suivant répertorie les certificats CA enregistrés dans votre AWS compte.

```
aws iot list-ca-certificates
```

Sortie :

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
            "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
            "status": "INACTIVE",
            "creationDate": 1569365372.053
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Utiliser votre propre certificat dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, voir ListCaCertificatesla section Référence des AWS CLI commandes.

`list-certificates-by-ca`

L'exemple de code suivant montre comment utiliser`list-certificates-by-ca`.

AWS CLI

Pour répertorier tous les certificats d'appareils signés avec un certificat CA

L'`list-certificates-by-ca`exemple suivant répertorie tous les certificats d'appareils de votre AWS compte qui sont signés avec le certificat CA spécifié.

```
aws iot list-certificates-by-ca \
    --ca-certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467
```

Sortie :

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "status": "ACTIVE",
            "creationDate": 1569363250.557
        }
    ]
}
```

Pour plus d'informations, consultez ListCertificatesByCA dans le manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, voir ListCertificatesByCala section Référence des AWS CLI commandes.

`list-certificates`

L'exemple de code suivant montre comment utiliser`list-certificates`.

AWS CLI

Exemple 1 : Pour répertorier les certificats enregistrés dans votre AWS compte

L'`list-certificates`exemple suivant répertorie tous les certificats enregistrés dans votre compte. Si vous avez dépassé la limite de pagination par défaut de 25, vous pouvez utiliser la valeur de `nextMarker` réponse de cette commande et la fournir à la commande suivante pour obtenir le prochain lot de résultats. Répétez jusqu'à ce qu'il `nextMarker` revienne sans valeur.

```
aws iot list-certificates
```

Sortie :

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
            "certificateId":
 "604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
            "status": "ACTIVE",
            "creationDate": 1556810537.617
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
            "certificateId":
 "262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
            "status": "ACTIVE",
            "creationDate": 1546447050.885
        },
        {
            "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/
b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
```

```
            "certificateId":
 "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
            "status": "ACTIVE",
            "creationDate": 1546292258.322
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
            "certificateId":
 "7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
            "status": "ACTIVE",
            "creationDate": 1541457693.453
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
            "certificateId":
 "54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
            "status": "ACTIVE",
            "creationDate": 1541113568.611
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
            "certificateId":
 "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
            "status": "ACTIVE",
            "creationDate": 1541022751.983
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListCertificatesla section Référence des AWS CLI commandes.

`list-custom-metrics`

L'exemple de code suivant montre comment utiliser`list-custom-metrics`.

AWS CLI

Pour répertorier vos statistiques personnalisées

L'`list-custom-metrics`exemple suivant répertorie toutes vos mesures personnalisées.

```
aws iot list-custom-metrics \
    --region us-east-1
```

Sortie :

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

Pour plus d'informations, consultez la section Mesures personnalisées dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous ListCustomMetricsà la section Référence des AWS CLI commandes.

`list-dimensions`

L'exemple de code suivant montre comment utiliser`list-dimensions`.

AWS CLI

Pour répertorier les dimensions de votre AWS compte

L'`list-dimensions`exemple suivant répertorie toutes les dimensions d' AWS IoT Device Defender définies dans votre AWS compte.

```
aws iot list-dimensions
```

Sortie :

```
{
    "dimensionNames": [
        "TopicFilterForAuthMessages",
        "TopicFilterForActivityMessages"
    ]
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListDimensionsà la section Référence des AWS CLI commandes.

`list-domain-configurations`

L'exemple de code suivant montre comment utiliser`list-domain-configurations`.

AWS CLI

Pour répertorier les configurations de domaine

L'`list-domain-configurations`exemple suivant répertorie les configurations de domaine de votre AWS compte qui ont le type de service spécifié.

```
aws iot list-domain-configurations \
    --service-type "DATA"
```

Sortie :

```
{
    "domainConfigurations":
    [
        {
            "domainConfigurationName": "additionalDataDomain",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh",
            "serviceType": "DATA"
        },

        {
            "domainConfigurationName": "iot:Jobs",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:Jobs",
            "serviceType": "JOBS"
        },
        {
            "domainConfigurationName": "iot:Data-ATS",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:Data-ATS",
```

```
            "serviceType": "DATA"
        },
        {
            "domainConfigurationName": "iot:CredentialProvider",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:CredentialProvider",
            "serviceType": "CREDENTIAL_PROVIDER"
        }
    ]
}
```

Pour plus d'informations, consultez la section Points de terminaison configurables dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListDomainConfigurationsà la section Référence des AWS CLI commandes.

`list-indices`

L'exemple de code suivant montre comment utiliser`list-indices`.

AWS CLI

Pour répertorier les indices de recherche configurés

L'`list-indices`exemple suivant répertorie tous les index de recherche configurés dans votre AWS compte. Si vous n'avez pas activé l'indexation des objets, il se peut que vous n'ayez aucun index.

```
aws iot list-indices
```

Sortie :

```
{
    "indexNames": [
        "AWS_Things"
    ]
}
```

Pour plus d'informations, consultez la section Managing Thing Indexing dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListIndicesà la section Référence des AWS CLI commandes.

`list-job-executions-for-job`

L'exemple de code suivant montre comment utiliser`list-job-executions-for-job`.

AWS CLI

Pour répertorier les offres d'emploi de votre AWS compte

L'`list-job-executions-for-job`exemple suivant répertorie toutes les exécutions de tâches pour une tâche de votre AWS compte, spécifiée par le JobID.

```
aws iot list-job-executions-for-job \
    --job-id my-ota-job
```

Sortie :

```
{
     "executionSummaries": [
         {
             "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/my_thing",
             "jobExecutionSummary": {
                 "status": "QUEUED",
                 "queuedAt": "2022-03-07T15:58:42.195000-08:00",
                 "lastUpdatedAt": "2022-03-07T15:58:42.195000-08:00",
                 "executionNumber": 1,
                 "retryAttempt": 0
             }
         }
     ]
 }
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListJobExecutionsForJobà la section Référence des AWS CLI commandes.

`list-job-executions-for-thing`

L'exemple de code suivant montre comment utiliser`list-job-executions-for-thing`.

AWS CLI

Pour répertorier les tâches exécutées pour un objet

L'`list-job-executions-for-thing`exemple suivant répertorie toutes les tâches exécutées pour l'objet nomméMyRaspberryPi.

```
aws iot list-job-executions-for-thing \
    --thing-name "MyRaspberryPi"
```

Sortie :

```
{
    "executionSummaries": [
        {
            "jobId": "example-job-01",
            "jobExecutionSummary": {
                "status": "QUEUED",
                "queuedAt": 1560787023.636,
                "lastUpdatedAt": 1560787023.636,
                "executionNumber": 1
            }
        }
    ]
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListJobExecutionsForThingà la section Référence des AWS CLI commandes.

`list-jobs`

L'exemple de code suivant montre comment utiliser`list-jobs`.

AWS CLI

Pour répertorier les offres d'emploi de votre AWS compte

L'`list-jobs`exemple suivant répertorie toutes les tâches de votre AWS compte, triées en fonction de leur statut.

```
aws iot list-jobs
```

Sortie :

```
{
    "jobs": [
        {
            "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
            "jobId": "example-job-01",
            "targetSelection": "SNAPSHOT",
            "status": "IN_PROGRESS",
            "createdAt": 1560787022.733,
            "lastUpdatedAt": 1560787026.294
        }
    ]
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListJobsà la section Référence des AWS CLI commandes.

`list-mitigation-actions`

L'exemple de code suivant montre comment utiliser`list-mitigation-actions`.

AWS CLI

Pour répertorier toutes les mesures d'atténuation définies

L'`list-mitigation-actions`exemple suivant répertorie toutes les mesures d'atténuation définies pour votre AWS compte et votre région. Pour chaque action, le nom, l'ARN et la date de création sont répertoriés.

```
aws iot list-mitigation-actions
```

Sortie :

```
{
    "actionIdentifiers": [
        {
            "actionName": "DeactivateCACertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
DeactivateCACertAction",
            "creationDate": "2019-12-10T11:12:47.574000-08:00"
        },
        {
            "actionName": "ResetPolicyVersionAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
ResetPolicyVersionAction",
            "creationDate": "2019-12-10T11:11:48.920000-08:00"
        },
        {
            "actionName": "PublishFindingToSNSAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
PublishFindingToSNSAction",
            "creationDate": "2019-12-10T11:10:49.546000-08:00"
        },
        {
            "actionName": "AddThingsToQuarantineGroupAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
            "creationDate": "2019-12-10T11:09:35.999000-08:00"
        },
        {
            "actionName": "UpdateDeviceCertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
UpdateDeviceCertAction",
            "creationDate": "2019-12-10T11:08:44.263000-08:00"
        },
        {
            "actionName": "SampleMitigationAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
SampleMitigationAction",
            "creationDate": "2019-12-10T11:03:41.840000-08:00"
        }
    ]
}
```


Pour plus d'informations, voir ListMitigationActions (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListMitigationActionsà la section Référence des AWS CLI commandes.

`list-mitigations-actions`

L'exemple de code suivant montre comment utiliser`list-mitigations-actions`.

AWS CLI

Pour répertorier toutes les mesures d'atténuation définies

L'`list-mitigations-actions`exemple suivant répertorie toutes les mesures d'atténuation définies pour votre AWS compte et votre région. Pour chaque action, le nom, l'ARN et la date de création sont répertoriés.

```
aws iot list-mitigation-actions
```

Sortie :

```
{
    "actionIdentifiers": [
        {
            "actionName": "DeactivateCACertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
DeactivateCACertAction",
            "creationDate": "2019-12-10T11:12:47.574000-08:00"
        },
        {
            "actionName": "ResetPolicyVersionAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
ResetPolicyVersionAction",
            "creationDate": "2019-12-10T11:11:48.920000-08:00"
        },
        {
            "actionName": "PublishFindingToSNSAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
PublishFindingToSNSAction",
            "creationDate": "2019-12-10T11:10:49.546000-08:00"
        },
        {
            "actionName": "AddThingsToQuarantineGroupAction",
```

```
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
            "creationDate": "2019-12-10T11:09:35.999000-08:00"
        },
        {
            "actionName": "UpdateDeviceCertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
UpdateDeviceCertAction",
            "creationDate": "2019-12-10T11:08:44.263000-08:00"
        },
        {
            "actionName": "SampleMitigationAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
SampleMitigationAction",
            "creationDate": "2019-12-10T11:03:41.840000-08:00"
        }
    ]
}
```

Pour plus d'informations, voir ListMitigationActions (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListMitigationsActionsà la section Référence des AWS CLI commandes.

`list-ota-updates`

L'exemple de code suivant montre comment utiliser`list-ota-updates`.

AWS CLI

Pour répertorier les mises à jour OTA pour le compte

L'`list-ota-updates`exemple suivant répertorie les mises à jour OTA disponibles.

```
aws iot list-ota-updates
```

Sortie :

```
{
    "otaUpdates": [
        {
```

```
            "otaUpdateId": "itsaupdate",
            "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/
itsaupdate",
            "creationDate": 1557863215.995
        }
    ]
}
```

Pour plus d'informations, consultez la section Liste OTAUpdates dans le guide de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListOtaUpdatesà la section Référence des AWS CLI commandes.

`list-outgoing-certificates`

L'exemple de code suivant montre comment utiliser`list-outgoing-certificates`.

AWS CLI

Pour répertorier les certificats transférés vers un autre AWS compte

L'`list-outgoing-certificates`exemple suivant répertorie tous les certificats de périphérique en cours de transfert vers un autre AWS compte à l'aide de la `transfer-certificate` commande.

```
aws iot list-outgoing-certificates
```

Sortie :

```
{
    "outgoingCertificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "transferredTo": "030714055129",
            "transferDate": 1569427780.441,
            "creationDate": 1569363250.557
        }
    ]
```

```
}
```

Pour plus d'informations, consultez ListOutgoingCertificatesle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListOutgoingCertificatesà la section Référence des AWS CLI commandes.

`list-policies`

L'exemple de code suivant montre comment utiliser`list-policies`.

AWS CLI

Pour répertorier les politiques définies dans votre AWS compte

L'`list-policies`exemple suivant répertorie toutes les politiques définies dans votre AWS compte.

```
aws iot list-policies
```

Sortie :

```
{
    "policies": [
        {
            "policyName": "UpdateDeviceCertPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
UpdateDeviceCertPolicy"
        },
        {
            "policyName": "PlantIoTPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/PlantIoTPolicy"
        },
        {
            "policyName": "MyPiGroup_Core-policy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/MyPiGroup_Core-
policy"
        }
    ]
}
```

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListPoliciesà la section Référence des AWS CLI commandes.

`list-policy-versions`

L'exemple de code suivant montre comment utiliser`list-policy-versions`.

AWS CLI

Exemple 1 : pour voir toutes les versions d'une politique

L'`list-policy-versions`exemple suivant répertorie toutes les versions de la politique spécifiée et leurs dates de création.

```
aws iot list-policy-versions \
    --policy-name LightBulbPolicy
```

Sortie :

```
{
    "policyVersions": [
        {
            "versionId": "2",
            "isDefaultVersion": true,
            "createDate": 1559925941.924
        },
        {
            "versionId": "1",
            "isDefaultVersion": false,
            "createDate": 1559925941.924
        }
    ]
}
```

Pour plus d'informations, consultez la section Politiques relatives à l'AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListPolicyVersionsà la section Référence des AWS CLI commandes.

`list-principal-things`

L'exemple de code suivant montre comment utiliser`list-principal-things`.

AWS CLI

Pour répertorier les objets attachés à un principal

L'`list-principal-things`exemple suivant répertorie les éléments attachés au principal spécifié par un ARN.

```
aws iot list-principal-things \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a351597670552 78e8
```

Sortie :

```
{
    "things": [
        "DeskLamp",
        "TableLamp"
    ]
}
```

Pour plus d'informations, consultez ListPrincipalThingsle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListPrincipalThingsà la section Référence des AWS CLI commandes.

`list-provisioning-template-versions`

L'exemple de code suivant montre comment utiliser`list-provisioning-template-versions`.

AWS CLI

Pour répertorier les versions des modèles de provisionnement

L'`list-provisioning-template-versions`exemple suivant répertorie les versions disponibles du modèle de provisionnement spécifié.

```
aws iot list-provisioning-template-versions \
```

```
    --template-name "widget-template"
```

Sortie :

```
{
    "versions": [
        {
            "versionId": 1,
            "creationDate": 1574800471.339,
            "isDefaultVersion": true
        },
        {
            "versionId": 2,
            "creationDate": 1574801192.317,
            "isDefaultVersion": false
        }
    ]
}
```

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous ListProvisioningTemplateVersionsà la section Référence des AWS CLI commandes.

`list-provisioning-templates`

L'exemple de code suivant montre comment utiliser`list-provisioning-templates`.

AWS CLI

Pour répertorier les modèles de provisionnement

L'`list-provisioning-templates`exemple suivant répertorie tous les modèles de provisionnement de votre AWS compte.

```
aws iot list-provisioning-templates
```

Sortie :

```
{
```

```
    "templates": [
        {
            "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/
widget-template",
            "templateName": "widget-template",
            "description": "A provisioning template for widgets",
            "creationDate": 1574800471.367,
            "lastModifiedDate": 1574801192.324,
            "enabled": false
        }
    ]
}
```

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous ListProvisioningTemplatesà la section Référence des AWS CLI commandes.

`list-role-aliases`

L'exemple de code suivant montre comment utiliser`list-role-aliases`.

AWS CLI

Pour répertorier les alias de rôle AWS IoT de votre compte AWS

L'`list-role-aliases`exemple suivant répertorie les alias de rôle AWS IoT de votre AWS compte.

```
aws iot list-role-aliases
```

Sortie :

```
{
    "roleAliases": [
        "ResidentAlias",
        "ElectricianAlias"
    ]
}
```

Pour plus d'informations, consultez ListRoleAliasesle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListRoleAliasesà la section Référence des AWS CLI commandes.

`list-scheduled-audits`

L'exemple de code suivant montre comment utiliser`list-scheduled-audits`.

AWS CLI

Pour répertorier les audits planifiés pour votre AWS compte

L'`list-scheduled-audits`exemple suivant répertorie tous les audits planifiés pour votre AWS compte.

```
aws iot list-scheduled-audits
```

Sortie :

```
{
    "scheduledAudits": [
        {
            "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
            "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSIoTDeviceDefenderDailyAudit",
            "frequency": "DAILY"
        },
        {
            "scheduledAuditName": "AWSDeviceDefenderWeeklyAudit",
            "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSDeviceDefenderWeeklyAudit",
            "frequency": "WEEKLY",
            "dayOfWeek": "SUN"
        }
    ]
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListScheduledAuditsà la section Référence des AWS CLI commandes.

`list-security-profiles-for-target`

L'exemple de code suivant montre comment utiliser`list-security-profiles-for-target`.

AWS CLI

Pour répertorier les profils de sécurité associés à une cible

L'`list-security-profiles-for-target`exemple suivant répertorie les profils de sécurité AWS IoT Device Defender associés à des appareils non enregistrés.

```
aws iot list-security-profiles-for-target  \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
unregistered-things"
```

Sortie :

```
{
    "securityProfileTargetMappings": [
        {
            "securityProfileIdentifier": {
                "name": "Testprofile",
                "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
Testprofile"
            },
            "target": {
                "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListSecurityProfilesForTargetà la section Référence des AWS CLI commandes.

`list-security-profiles`

L'exemple de code suivant montre comment utiliser`list-security-profiles`.

AWS CLI

Pour répertorier les profils de sécurité de votre AWS compte

L'`list-security-profiles`exemple suivant répertorie tous les profils de sécurité AWS IoT Device Defender définis dans votre AWS compte.

```
aws iot list-security-profiles
```

Sortie :

```
{
    "securityProfileIdentifiers": [
        {
            "name": "Testprofile",
            "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
        }
    ]
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListSecurityProfilesà la section Référence des AWS CLI commandes.

`list-streams`

L'exemple de code suivant montre comment utiliser`list-streams`.

AWS CLI

Pour répertorier les streams du compte

L'`list-streams`exemple suivant répertorie tous les streams de votre AWS compte.

```
aws iot list-streams
```

Sortie :

```
{
```

```
    "streams": [
        {
            "streamId": "stream12345",
            "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
            "streamVersion": 1,
            "description": "This stream is used for Amazon FreeRTOS OTA Update
 12345."
        },
        {
            "streamId": "stream54321",
            "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream54321",
            "streamVersion": 1,
            "description": "This stream is used for Amazon FreeRTOS OTA Update
 54321."
        }
    ]
}
```

Pour plus d'informations, consultez ListStreamsle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListStreamsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour afficher les balises et leurs valeurs associées à une ressource

L'`list-tags-for-resource`exemple suivant affiche les balises et les valeurs associées au groupe d'objets`LightBulbs`.

```
aws iot list-tags-for-resource \
    --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Sortie :

```
{
    "tags": [
```

```
        {
            "Key": "Assembly",
            "Value": "Fact1NW"
        },
        {
            "Key": "MyTag",
            "Value": "777"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage de vos ressources AWS IoT dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-targets-for-policy`

L'exemple de code suivant montre comment utiliser`list-targets-for-policy`.

AWS CLI

Pour répertorier les principes associés à une politique AWS IoT

L'`list-targets-for-policy`exemple suivant répertorie les certificats de périphérique auxquels la politique spécifiée est attachée.

```
aws iot list-targets-for-policy \
    --policy-name UpdateDeviceCertPolicy
```

Sortie :

```
{
    "targets": [
        "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
        "arn:aws:iot:us-west-2:123456789012:cert/
d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"
    ]
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTargetsForPolicyà la section Référence des AWS CLI commandes.

`list-targets-for-security-profile`

L'exemple de code suivant montre comment utiliser`list-targets-for-security-profile`.

AWS CLI

Pour répertorier les cibles auxquelles un profil de sécurité est appliqué

L'`list-targets-for-security-profile`exemple suivant répertorie les cibles auxquelles le profil de sécurité AWS IoT Device Defender nommé `PossibleIssue` est appliqué.

```
aws iot list-targets-for-security-profile \
    --security-profile-name Testprofile
```

Sortie :

```
{
    "securityProfileTargets": [
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
        },
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"
        }
    ]
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTargetsForSecurityProfileà la section Référence des AWS CLI commandes.

`list-thing-groups-for-thing`

L'exemple de code suivant montre comment utiliser`list-thing-groups-for-thing`.

AWS CLI

Pour répertorier les groupes auxquels appartient un objet

L'`list-thing-groups-for-thing`exemple suivant répertorie les groupes auxquels appartient l'objet spécifié.

```
aws iot list-thing-groups-for-thing \
    --thing-name MyLightBulb
```

Sortie :

```
{
    "thingGroups": [
        {
            "groupName": "DeadBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/DeadBulbs"
        },
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        }
    ]
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingGroupsForThingà la section Référence des AWS CLI commandes.

`list-thing-groups`

L'exemple de code suivant montre comment utiliser`list-thing-groups`.

AWS CLI

Pour répertorier les groupes d'objets définis dans votre AWS compte

L'`describe-thing-group`exemple suivant répertorie tous les groupes d'objets définis dans votre AWS compte.

```
aws iot list-thing-groups
```

Sortie :

```
{
    "thingGroups": [
        {
            "groupName": "HalogenBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs"
        },
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        }
    ]
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingGroupsà la section Référence des AWS CLI commandes.

`list-thing-principals`

L'exemple de code suivant montre comment utiliser`list-thing-principals`.

AWS CLI

Pour répertorier les principes associés à un objet

L'`list-thing-principals`exemple suivant répertorie les principaux (certificats X.509, utilisateurs IAM, groupes, rôles, identités Amazon Cognito ou identités fédérées) associés à l'objet spécifié.

```
aws iot list-thing-principals \
    --thing-name MyRaspberryPi
```

Sortie :

```
{
```

```
    "principals": [
        "arn:aws:iot:us-
west-2:123456789012:cert/33475ac865079a5ffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"
    ]
}
```

Pour plus d'informations, consultez ListThingPrincipalsle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingPrincipalsà la section Référence des AWS CLI commandes.

`list-thing-types`

L'exemple de code suivant montre comment utiliser`list-thing-types`.

AWS CLI

Pour répertorier les types d'objets définis

L'`list-thing-types`exemple suivant affiche une liste des types d'objets définis dans votre AWS compte.

```
aws iot list-thing-types
```

Sortie :

```
{
    "thingTypes": [
        {
            "thingTypeName": "LightBulb",
            "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/
LightBulb",
            "thingTypeProperties": {
                "thingTypeDescription": "light bulb type",
                "searchableAttributes": [
                    "model",
                    "wattage"
                ]
            },
            "thingTypeMetadata": {
            "deprecated": false,
            "creationDate": 1559772562.498
            }
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingTypesà la section Référence des AWS CLI commandes.

`list-things-in-billing-group`

L'exemple de code suivant montre comment utiliser`list-things-in-billing-group`.

AWS CLI

Pour répertorier les éléments d'un groupe de facturation

L'`list-things-in-billing-group`exemple suivant répertorie les éléments qui se trouvent dans le groupe de facturation spécifié.

```
aws iot list-things-in-billing-group \
    --billing-group-name GroupOne
```

Sortie :

```
{
    "things": [
        "MyOtherLightBulb",
        "MyLightBulb"
    ]
}
```

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingsInBillingGroupà la section Référence des AWS CLI commandes.

`list-things-in-thing-group`

L'exemple de code suivant montre comment utiliser`list-things-in-thing-group`.

AWS CLI

Pour répertorier les éléments appartenant à un groupe

L'`list-things-in-thing-group`exemple suivant répertorie les objets appartenant au groupe d'objets spécifié.

```
aws iot list-things-in-thing-group \
    --thing-group-name LightBulbs
```

Sortie :

```
{
    "things": [
        "MyLightBulb"
    ]
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingsInThingGroupà la section Référence des AWS CLI commandes.

`list-things`

L'exemple de code suivant montre comment utiliser`list-things`.

AWS CLI

Exemple 1 : pour répertorier tous les éléments du registre

L'`list-things`exemple suivant répertorie les objets (appareils) définis dans le registre AWS IoT de votre AWS compte.

```
aws iot list-things
```

Sortie :

```
{
    "things": [
```

```
        {
            "thingName": "ThirdBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/ThirdBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 2
        },
        {
            "thingName": "MyOtherLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3
        },
        {
            "thingName": "MyLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1
        },
        {
        "thingName": "SampleIoTThing",
        "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/SampleIoTThing",
        "attributes": {},
        "version": 1
        }
    ]
}
```

Exemple 2 : pour répertorier les éléments définis dotés d'un attribut spécifique

L'`list-things`exemple suivant affiche une liste d'objets dotés d'un attribut nomméwattage.

```
aws iot list-things \
```

```
    --attribute-name wattage
```

Sortie :

```
{
    "things": [
        {
            "thingName": "MyLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1
        },
        {
            "thingName": "MyOtherLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3
        }
    ]
}
```

Pour plus d'informations, consultez la section Comment gérer les objets avec le registre dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListThingsà la section Référence des AWS CLI commandes.

`list-topic-rule-destinations`

L'exemple de code suivant montre comment utiliser`list-topic-rule-destinations`.

AWS CLI

Pour répertorier votre sujet, réglez les destinations

L'`list-topic-rule-destinations`exemple suivant répertorie toutes les destinations des règles de rubrique que vous avez définies dans la AWS région actuelle.

```
aws iot list-topic-rule-destinations
```

Sortie :

```
{
    "destinationSummaries": [
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "status": "ENABLED",
            "httpUrlSummary": {
                "confirmationUrl": "https://example.com"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des destinations de règles thématiques dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTopicRuleDestinationsà la section Référence des AWS CLI commandes.

`list-topic-rules`

L'exemple de code suivant montre comment utiliser`list-topic-rules`.

AWS CLI

Pour répertorier vos règles

L'`list-topic-rules`exemple suivant répertorie toutes les règles que vous avez définies.

```
aws iot list-topic-rules
```

Sortie :

```
{
    "rules": [
```

```
        {
            "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyRPiLowMoistureAlertRule",
            "ruleName": "MyRPiLowMoistureAlertRule",
            "topicPattern": "$aws/things/MyRPi/shadow/update/accepted",
            "createdAt": 1558624363.0,
            "ruleDisabled": false
        },
        {
            "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyPlantPiMoistureAlertRule",
            "ruleName": "MyPlantPiMoistureAlertRule",
            "topicPattern": "$aws/things/MyPlantPi/shadow/update/accepted",
            "createdAt": 1541458459.0,
            "ruleDisabled": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation de vos règles dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTopicRulesà la section Référence des AWS CLI commandes.

`list-v2-logging-levels`

L'exemple de code suivant montre comment utiliser`list-v2-logging-levels`.

AWS CLI

Pour répertorier les niveaux de journalisation

L'`list-v2-logging-levels`exemple suivant répertorie les niveaux de journalisation configurés. Si les niveaux de journalisation n'ont pas été définis, un `NotConfiguredException` se produit lorsque vous exécutez cette commande.

```
aws iot list-v2-logging-levels
```

Sortie :

```
{
```

```
    "logTargetConfigurations": [
        {
            "logTarget": {
                "targetType": "DEFAULT"
            },
            "logLevel": "ERROR"
        }
    ]
}
```

- Pour plus de détails sur l'API, consultez ListV2 LoggingLevels dans AWS CLI Command Reference.

`list-violation-events`

L'exemple de code suivant montre comment utiliser`list-violation-events`.

AWS CLI

Pour répertorier les violations du profil de sécurité au cours d'une période donnée

L'`list-violation-events`exemple suivant répertorie les violations survenues entre le 5 juin 2019 et le 12 juin 2019 pour tous les profils de sécurité AWS IoT Device Defender du AWS compte courant et de AWS la région.

```
aws iot list-violation-events \
    --start-time 1559747125 \
    --end-time 1560351925
```

Sortie :

```
{
    "violationEvents": [
        {
            "violationId": "174db59167fa474c80a652ad1583fd44",
            "thingName": "iotconsole-1560269126751-1",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
```

```
                "value": {
                    "count": 10
                },
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        "metricValue": {
            "count": 0
        },
        "violationEventType": "in-alarm",
        "violationEventTime": 1560279000.0
    },
    {
        "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
        "thingName": "TvnQoEoU",
        "securityProfileName": "Testprofile",
        "behavior": {
            "name": "CellularBandwidth",
            "metric": "aws:message-byte-size",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 128
                },
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        "metricValue": {
            "count": 110
        },
        "violationEventType": "in-alarm",
        "violationEventTime": 1560276600.0
    },
    {
        "violationId": "74aa393adea02e6648f3ac362beed55e",
        "thingName": "iotconsole-1560269232412-2",
        "securityProfileName": "Testprofile",
        "behavior": {
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
```

```
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560276600.0
        },
        {
            "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560276600.0
        }
    ]
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListViolationEventsà la section Référence des AWS CLI commandes.

`register-ca-certificate`

L'exemple de code suivant montre comment utiliser`register-ca-certificate`.

AWS CLI

Pour enregistrer un certificat d'autorité de certification (CA)

L'`register-ca-certificate`exemple suivant enregistre un certificat CA. La commande fournit le certificat CA et un certificat de vérification de clé qui prouve que vous possédez la clé privée associée au certificat CA.

```
aws iot register-ca-certificate \
    --ca-certificate file://rootCA.pem \
    --verification-cert file://verificationCert.pem
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
    "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467"
 }
```

Pour plus d'informations, consultez la section S'inscrire CACertificate dans le guide de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous RegisterCaCertificateà la section Référence des AWS CLI commandes.

`register-certificate`

L'exemple de code suivant montre comment utiliser`register-certificate`.

AWS CLI

Pour enregistrer un certificat d'appareil auto-signé

L'`register-certificate`exemple suivant enregistre le certificat de `deviceCert.pem` périphérique signé par le certificat `rootCA.pem` CA. Le certificat CA doit être enregistré avant que vous ne l'utilisiez pour enregistrer un certificat d'appareil auto-signé. Le certificat auto-signé doit être signé par le même certificat CA que celui que vous avez transmis à cette commande.

```
aws iot register-certificate \
    --certificate-pem file://deviceCert.pem \
    --ca-certificate-pem file://rootCA.pem
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
    "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
 }
```

Pour plus d'informations, consultez RegisterCertificatele manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous RegisterCertificateà la section Référence des AWS CLI commandes.

`register-thing`

L'exemple de code suivant montre comment utiliser`register-thing`.

AWS CLI

Pour enregistrer un objet

L'`register-thing`exemple suivant enregistre un objet à l'aide d'un modèle de provisioning.

```
aws iot register-thing \
    --template-body '{"Parameters":{"ThingName":
{"Type":"String"},"AWS::IoT::Certificate::Id":{"Type":"String"}},"Resources":
 {"certificate":{"Properties":{"CertificateId":
{"Ref":"AWS::IoT::Certificate::Id"},"Status":"Active"},"Type":"AWS::IoT::Certificate"},"poli
{"Properties":{"PolicyName":"MyIotPolicy"},"Type":"AWS::IoT::Policy"},"thing":
{"OverrideSettings":
```

```
{"AttributePayload":"MERGE","ThingGroups":"DO_NOTHING","ThingTypeName":"REPLACE"},"Propertie
{"AttributePayload":{},"ThingGroups":[],"ThingName":
{"Ref":"ThingName"},"ThingTypeName":"VirtualThings"},"Type":"AWS::IoT::Thing"}}}' \
    --parameters '{"ThingName":"Register-thing-
trial-1","AWS::IoT::Certificate::Id":"799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e3
```

Sortie :

```
{
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgIUYLk81I35cIppobpw
HiOJ2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi
5jb20g\nSW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIwMDcyMzE2NDUw
\nOVoXDTQ5MTIzMT
IzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0\nZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAO71uADhdBajqTmgrMV5\nmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMAckeX8KeblIOTKzORI
RDXnyE
\n6lVOwjgAsd0ku22rFxex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nKqiuBJVNOGKTcQ
Hd1RNOrddwwu6kFJLeKDmEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nGXu7u0W3hn6nlRN8qVoka0uW12p53xM7oHVz
Gf+cxKBxlbOhGkp6yCfTSkUBm3Sp\n9zLw35kiHXVm4EVpwgNlnk6XcIGIkw8a/iy4pzmvuGAANY1/uU/
zgCjymw
ZT5S30\nBV0CAwEAAaNgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR0OBBYEFOVtvZ
9Aj2RYFnkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nAQH/
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IB
AQCXCQcpOtubS5ftOsDMTcpP/jNX
\nDHyArxmjpSc2aCdmm7WX59lTKWyAdxGAvqaDVWqTo0oXI7tZ8w7aINlGi5
pXnifx\n3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fE7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4srOCKynS5wMJuQZlHQOnabVwnwK4Y0Mflp
\np9+4susFUR9aT3BT1AcIwqSpzhlKhh4Iz7ND
kRn4amsUT210jg/zOO1Ow+BTHcVQ\nJly8XDu0CWSu04q6SnaBzHmlySIajxuRTP/AdfRouP1OXe
+qlbPOBcvVvF
8o\n-----END CERTIFICATE-----\n",
    "resourceArns": {
        "certificate": "arn:aws:iot:us-
west-2:571032923833:cert/799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e30404b9233c",
        "thing": "arn:aws:iot:us-west-2:571032923833:thing/Register-thing-trial-1"
    }
}
```

Pour plus d'informations, consultez la section Provisionnement par un utilisateur de confiance dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous RegisterThingà la section Référence des AWS CLI commandes.

`reject-certificate-transfer`

L'exemple de code suivant montre comment utiliser`reject-certificate-transfer`.

AWS CLI

Pour refuser un transfert de certificat

L'`reject-certificate-transfer`exemple suivant rejette le transfert du certificat d'appareil spécifié depuis un autre AWS compte.

```
aws iot reject-certificate-transfer \
    --certificate-
id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Transférer un certificat vers un autre compte dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous RejectCertificateTransferà la section Référence des AWS CLI commandes.

`remove-thing-from-billing-group`

L'exemple de code suivant montre comment utiliser`remove-thing-from-billing-group`.

AWS CLI

Pour supprimer un élément d'un groupe de facturation

L'`remove-thing-from-billing-group`exemple suivant supprime l'élément spécifié d'un groupe de facturation.

```
aws iot remove-thing-from-billing-group \
    --billing-group-name GroupOne \
```

```
    --thing-name MyOtherLightBulb
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous RemoveThingFromBillingGroupà la section Référence des AWS CLI commandes.

`remove-thing-from-thing-group`

L'exemple de code suivant montre comment utiliser`remove-thing-from-thing-group`.

AWS CLI

Pour supprimer un objet d'un groupe d'objets

L'`remove-thing-from-thing-group`exemple suivant supprime l'objet spécifié d'un groupe d'objets.

```
aws iot remove-thing-from-thing-group \
    --thing-name bulb7 \
    --thing-group-name DeadBulbs
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Thing Groups < https://docs.aws.amazon.com/iot/latest/developerguide/thing -groups.html > dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous RemoveThingFromThingGroupà la section Référence des AWS CLI commandes.

`replace-topic-rule`

L'exemple de code suivant montre comment utiliser`replace-topic-rule`.

AWS CLI

Pour mettre à jour la définition de règle d'une rubrique

L'`replace-topic-rule`exemple suivant met à jour la règle spécifiée pour envoyer une alerte SNS lorsque les relevés du niveau d'humidité du sol sont trop faibles.

```
aws iot replace-topic-rule \
    --rule-name MyRPiLowMoistureAlertRule \
    --topic-rule-payload "{\"sql\": \"SELECT * FROM '$aws/things/MyRPi/shadow/
update/accepted' WHERE state.reported.moisture = 'low'\", \"description\": \"Sends
 an alert when soil moisture level readings are too low.\",\"actions\": [{\"sns
\":{\"targetArn\":\"arn:aws:sns:us-west-2:123456789012:MyRPiLowMoistureTopic\",
\"roleArn\":\"arn:aws:iam::123456789012:role/service-role/MyRPiLowMoistureTopicRole
\",\"messageFormat\": \"RAW\"}}],\"ruleDisabled\": false,\"awsIotSqlVersion\":
\"2016-03-23\"}"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une règle AWS IoT dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ReplaceTopicRuleà la section Référence des AWS CLI commandes.

`search-index`

L'exemple de code suivant montre comment utiliser`search-index`.

AWS CLI

Pour interroger l'index des objets

L'`search-index`exemple suivant interroge l'`AWS_Things`index pour les objets dont le type est`LightBulb`.

```
aws iot search-index \
    --index-name "AWS_Things" \
    --query-string "thingTypeName:LightBulb"
```

Sortie :

```
{
    "things": [
        {
            "thingName": "MyLightBulb",
            "thingId": "40da2e73-c6af-406e-b415-15acae538797",
            "thingTypeName": "LightBulb",
```

```
            "thingGroupNames": [
                "LightBulbs",
                "DeadBulbs"
            ],
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "connectivity": {
                "connected": false
            }
        },
        {
            "thingName": "ThirdBulb",
            "thingId": "615c8455-33d5-40e8-95fd-3ee8b24490af",
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "connectivity": {
                "connected": false
            }
        },
        {
            "thingName": "MyOtherLightBulb",
            "thingId": "6dae0d3f-40c1-476a-80c4-1ed24ba6aa11",
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "connectivity": {
                "connected": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Managing Thing Indexing dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous SearchIndexà la section Référence des AWS CLI commandes.

`set-default-authorizer`

L'exemple de code suivant montre comment utiliser`set-default-authorizer`.

AWS CLI

Pour définir un autorisateur par défaut

L'`set-default-authorizer`exemple suivant définit l'autorisateur personnalisé `CustomAuthorizer` comme étant l'autorisateur par défaut.

```
aws iot set-default-authorizer \
    --authorizer-name CustomAuthorizer
```

Sortie :

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

Pour plus d'informations, consultez CreateDefaultAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous SetDefaultAuthorizerà la section Référence des AWS CLI commandes.

`set-default-policy-version`

L'exemple de code suivant montre comment utiliser`set-default-policy-version`.

AWS CLI

Pour définir la version par défaut d'une politique

L'`set-default-policy-version`exemple suivant définit la version par défaut sur 2 pour la politique nomméeUpdateDeviceCertPolicy.

```
aws iot set-default-policy-version \
```

```
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetDefaultPolicyVersionà la section Référence des AWS CLI commandes.

`set-v2-logging-level`

L'exemple de code suivant montre comment utiliser`set-v2-logging-level`.

AWS CLI

Pour définir le niveau de journalisation d'un groupe d'objets

L'`set-v2-logging-level`exemple suivant définit le niveau de journalisation pour consigner les avertissements pour le groupe d'objets spécifié.

```
aws iot set-v2-logging-level \
    --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \
    --log-level WARN
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, consultez SetV2 LoggingLevel dans AWS CLI Command Reference.

`set-v2-logging-options`

L'exemple de code suivant montre comment utiliser`set-v2-logging-options`.

AWS CLI

Pour définir les options de journalisation

L'`set-v2-logging-options`exemple suivant définit le niveau de verbosité de journalisation par défaut sur ERROR et indique l'ARN à utiliser pour la journalisation.

```
aws iot set-v2-logging-options \
    --default-log-level ERROR \
```

```
    --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, consultez SetV2 LoggingOptions dans AWS CLI Command Reference.

`start-audit-mitigation-actions-task`

L'exemple de code suivant montre comment utiliser`start-audit-mitigation-actions-task`.

AWS CLI

Appliquer une mesure d'atténuation aux résultats d'un audit

L'`start-audit-mitigation-actions-task`exemple suivant applique l'`ResetPolicyVersionAction`action (qui efface la politique) au résultat unique spécifié.

```
aws iot start-audit-mitigation-actions-task \
    --task-id "myActionsTaskId" \
    --target "findingIds=[\"0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464\"]" \
    --audit-check-to-actions-mapping
 "IOT_POLICY_OVERLY_PERMISSIVE_CHECK=[\"ResetPolicyVersionAction\"]" \
    --client-request-token "adhadhahda"
```

Sortie :

```
{
    "taskId": "myActionsTaskId"
}
```

Pour plus d'informations, voir StartAuditMitigationActionsTask (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous StartAuditMitigationActionsTaskà la section Référence des AWS CLI commandes.

`start-on-demand-audit-task`

L'exemple de code suivant montre comment utiliser`start-on-demand-audit-task`.

AWS CLI

Pour démarrer un audit immédiatement

L'`start-on-demand-audit-task`exemple suivant lance un audit AWS IoT Device Defender et effectue trois vérifications de certificats.

```
aws iot start-on-demand-audit-task \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Sortie :

```
{
    "taskId": "a3aea009955e501a31b764abe1bebd3d"
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous StartOnDemandAuditTaskà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour spécifier une clé de balise et une valeur pour une ressource

L'`tag-resource`exemple suivant applique la balise avec une clé `Assembly` et la valeur `Fact1NW` au groupe d'objets`LightBulbs`.

```
aws iot tag-resource \
    --tags Key=Assembly,Value="Fact1NW" \
    --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources AWS IoT dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`test-authorization`

L'exemple de code suivant montre comment utiliser`test-authorization`.

AWS CLI

Pour tester vos politiques en matière d' AWS IoT

L'`test-authorization`exemple suivant teste les politiques AWS IoT associées au principal spécifié.

```
aws iot test-authorization \
    --auth-infos actionType=CONNECT,resources=arn:aws:iot:us-
east-1:123456789012:client/client1 \
    --principal arn:aws:iot:us-west-2:123456789012:cert/
aab1068f7f43ac3e3cae4b3a8aa3f308d2a750e6350507962e32c1eb465d9775
```

Sortie :

```
{
    "authResults": [
        {
            "authInfo": {
                "actionType": "CONNECT",
                "resources": [
                    "arn:aws:iot:us-east-1:123456789012:client/client1"
                ]
            },
            "allowed": {
                "policies": [
                    {
                        "policyName": "TestPolicyAllowed",
                        "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyAllowed"
                    }
                ]
            },
```

```
            "denied": {
                "implicitDeny": {
                    "policies": [
                        {
                            "policyName": "TestPolicyDenied",
                            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyDenied"
                        }
                    ]
                },
                "explicitDeny": {
                    "policies": [
                        {
                            "policyName": "TestPolicyExplicitDenied",
                            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyExplicitDenied"
                        }
                    ]
                }
            },
            "authDecision": "IMPLICIT_DENY",
            "missingContextValues": []
        }
    ]
}
```

Pour plus d'informations, consultez TestAuthorizationle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous TestAuthorizationà la section Référence des AWS CLI commandes.

`test-invoke-authorizer`

L'exemple de code suivant montre comment utiliser`test-invoke-authorizer`.

AWS CLI

Pour tester votre autorisateur personnalisé

L'`test-invoke-authorizer`exemple suivant teste votre autorisateur personnalisé.

```
aws iot test-invoke-authorizer \
    --authorizer-name IoTAuthorizer \
```

```
    --token allow \
    --token-signature "mE0GvaHqy9nER/
FdgtJX5lXYEJ3b3vE7t1gEszc0TKGgLKWXTnPkb2AbKnOAZ8lGyoN5dVtWDWVmr25m7+
+zjbYIMk2TBvyGXhOmvKFBPkdgyA43KL6SiZy0cTqlPMcQDsP7VX2rXr7CTowCxSNKphGXdQe0/
I5dQ+JO6KUaHwCmupt0/MejKtaNwiia064j6wprOAUwG5S1IYFuRd0X
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUShEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcajxzGl02ExvyCUNctCV9dYlRRGJj
```

Sortie :

```
{
    "isAuthenticated": true,
    "principalId": "principalId",
    "policyDocuments": [
        "{"Version":"2012-10-17","Statement":
[{"Action":"iot:Publish","Effect":"Allow","Resource":"arn:aws:iot:us-
west-2:123456789012:topic/customauthtesting"}]}"
    ],
    "refreshAfterInSeconds": 600,
    "disconnectAfterInSeconds": 3600
}
```

Pour plus d'informations, consultez TestInvokeAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous TestInvokeAuthorizerà la section Référence des AWS CLI commandes.

`transfer-certificate`

L'exemple de code suivant montre comment utiliser`transfer-certificate`.

AWS CLI

Pour transférer un certificat d'appareil vers un autre AWS compte

L'`transfer-certificate`exemple suivant transfère un certificat d'appareil vers un autre AWS compte. Le certificat et le AWS compte sont identifiés par un identifiant.

```
aws iot transfer-certificate \
    --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \
    --target-aws-account 030714055129
```

Sortie :

```
{
    "transferredCertificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}
```

Pour plus d'informations, consultez la section Transférer un certificat vers un autre compte dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous TransferCertificateà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une clé de balise d'une ressource

L'`untag-resource`exemple suivant supprime la balise `MyTag` et sa valeur du groupe d'objets`LightBulbs`.

```
command
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources AWS IoT dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-account-audit-configuration`

L'exemple de code suivant montre comment utiliser`update-account-audit-configuration`.

AWS CLI

Exemple 1 : pour activer les notifications Amazon SNS pour les notifications d'audit

L'`update-account-audit-configuration`exemple suivant active les notifications Amazon SNS pour les notifications d'audit AWS IoT Device Defender, en spécifiant une cible et le rôle utilisé pour écrire sur cette cible.

```
aws iot update-account-audit-configuration \
    --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-west-2:123456789012:ddaudits\",roleArn=\"arn:aws:iam::123456789012:role/service-role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour activer un contrôle d'audit

L'`update-account-audit-configuration`exemple suivant active le contrôle d'audit AWS IoT Device Defender nomméAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK. Vous ne pouvez pas désactiver un contrôle d'audit s'il fait partie d'un ou de plusieurs audits planifiés pour le AWS compte. `targetCheckNames`

```
aws iot update-account-audit-configuration \
    --audit-check-configurations
 "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateAccountAuditConfigurationà la section Référence des AWS CLI commandes.

`update-audit-suppression`

L'exemple de code suivant montre comment utiliser`update-audit-suppression`.

AWS CLI

Pour mettre à jour la suppression d'un résultat d'audit

L'`update-audit-suppression`exemple suivant met à jour la date d'expiration de la suppression d'un résultat d'audit au 21/09/2020.

```
aws iot update-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId=c7691e<shortened> \
    --no-suppress-indefinitely \
    --expiration-date 2020-09-21
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Audit sur les suppressions dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateAuditSuppressionà la section Référence des AWS CLI commandes.

`update-authorizer`

L'exemple de code suivant montre comment utiliser`update-authorizer`.

AWS CLI

Pour mettre à jour un autorisateur personnalisé

L'`update-authorizer`exemple suivant montre l'état `CustomAuthorizer2` de deux`INACTIVE`.

```
aws iot update-authorizer \
    --authorizer-name CustomAuthorizer2 \
    --status INACTIVE
```

Sortie :

```
{
    "authorizerName": "CustomAuthorizer2",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

Pour plus d'informations, consultez UpdateAuthorizerle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateAuthorizerà la section Référence des AWS CLI commandes.

update-billing-group

L'exemple de code suivant montre comment utiliserupdate-billing-group.

AWS CLI

Pour mettre à jour les informations relatives à un groupe de facturation

L'update-billing-groupexemple suivant met à jour la description du groupe de facturation spécifié.

```
aws iot update-billing-group \
    --billing-group-name GroupOne \
    --billing-group-properties "billingGroupDescription=\"Primary bulb billing group
\""
```

Sortie :

```
{
    "version": 2
}
```

Pour plus d'informations, consultez la section Groupes de facturation dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateBillingGroupà la section Référence des AWS CLI commandes.

update-ca-certificate

L'exemple de code suivant montre comment utiliserupdate-ca-certificate.

AWS CLI

Pour mettre à jour un certificat d'autorité de certification (CA)

L'update-ca-certificateexemple suivant attribue au certificat CA spécifié le statut ACTIF.

```
aws iot update-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f05362952074264600 58cd1467 \
    --new-status ACTIVE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Mise CACertificate à jour de la référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateCaCertificateà la section Référence des AWS CLI commandes.

`update-certificate`

L'exemple de code suivant montre comment utiliser`update-certificate`.

AWS CLI

Pour mettre à jour le certificat d'un appareil

L'`update-certificate`exemple suivant définit le certificat d'appareil spécifié sur le statut INACTIF.

```
aws iot update-certificate \
    --certificate-
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \
    --new-status INACTIVE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UpdateCertificatele manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateCertificateà la section Référence des AWS CLI commandes.

`update-custom-metric`

L'exemple de code suivant montre comment utiliser`update-custom-metric`.

AWS CLI

Pour mettre à jour une métrique personnalisée

L'`update-custom-metric`exemple suivant met à jour une métrique personnalisée pour en ajouter une nouvelle`display-name`.

```
aws iot update-custom-metric \
    --metric-name batteryPercentage \
    --display-name 'remaining battery percentage on device' \
    --region us-east-1
```

Sortie :

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage",
    "metricType": "number",
    "displayName": "remaining battery percentage on device",
    "creationDate": "2020-11-17T23:01:35.110000-08:00",
    "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"
}
```

Pour plus d'informations, consultez la section Mesures personnalisées dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous UpdateCustomMetricà la section Référence des AWS CLI commandes.

`update-dimension`

L'exemple de code suivant montre comment utiliser`update-dimension`.

AWS CLI

Pour mettre à jour une dimension

L'`update-dimension`exemple suivant met à jour une dimension.

```
aws iot update-dimension \
    --name TopicFilterForAuthMessages \
    --string-values device/${iot:ClientId}/auth
```

Sortie :

```
{
    "name": "TopicFilterForAuthMessages",
```

```
    "lastModifiedDate": 1585866222.317,
    "stringValues": [
        "device/${iot:ClientId}/auth"
    ],
    "creationDate": 1585854500.474,
    "type": "TOPIC_FILTER",
    "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/
TopicFilterForAuthMessages"
}
```

Pour plus d'informations, consultez la section Délimitation des métriques dans les profils de sécurité à l'aide de dimensions dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous UpdateDimensionà la section Référence des AWS CLI commandes.

`update-domain-configuration`

L'exemple de code suivant montre comment utiliser`update-domain-configuration`.

AWS CLI

Pour mettre à jour la configuration d'un domaine

L'`update-domain-configuration`exemple suivant désactive la configuration de domaine spécifiée.

```
aws iot update-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --domain-configuration-status "DISABLED"
```

Sortie :

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

Pour plus d'informations, consultez la section Points de terminaison configurables dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainConfigurationà la section Référence des AWS CLI commandes.

`update-dynamic-thing-group`

L'exemple de code suivant montre comment utiliser`update-dynamic-thing-group`.

AWS CLI

Pour mettre à jour un groupe d'objets dynamique

L'`update-dynamic-thing-group`exemple suivant met à jour le groupe d'objets dynamiques spécifié. Il fournit une description et met à jour la chaîne de requête pour modifier les critères d'appartenance au groupe.

```
aws iot update-dynamic-thing-group \
    --thing-group-name "RoomTooWarm"
    --thing-group-properties "thingGroupDescription=\"This thing group contains
 rooms warmer than 65F.\"" \
    --query-string "attributes.temperature>65"
```

Sortie :

```
{
    "version": 2
}
```

Pour plus d'informations, consultez la section Dynamic Thing Groups dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateDynamicThingGroupà la section Référence des AWS CLI commandes.

`update-event-configurations`

L'exemple de code suivant montre comment utiliser`update-event-configurations`.

AWS CLI

Pour afficher les types d'événements publiés

L'`update-event-configurations`exemple suivant met à jour la configuration pour activer les messages lorsque le certificat CA est ajouté, mis à jour ou supprimé.

```
aws iot update-event-configurations \
    --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Messages d'événements dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateEventConfigurationsà la section Référence des AWS CLI commandes.

`update-indexing-configuration`

L'exemple de code suivant montre comment utiliser`update-indexing-configuration`.

AWS CLI

Pour activer l'indexation des objets

L'`update-indexing-configuration`exemple suivant active l'indexation des objets pour prendre en charge la recherche dans les données de registre, les données fictives et l'état de connectivité des objets à l'aide de l'index AWS_Things.

```
aws iot update-indexing-configuration
    --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Managing Thing Indexing dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateIndexingConfigurationà la section Référence des AWS CLI commandes.

`update-job`

L'exemple de code suivant montre comment utiliser`update-job`.

AWS CLI

Pour obtenir le statut détaillé d'une tâche

L'`update-job`exemple suivant permet d'obtenir le statut détaillé de la tâche dont l'ID est`example-job-01`.

```
aws iot describe-job \
    --job-id "example-job-01"
```

Sortie :

```
{
    "job": {
        "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
        "jobId": "example-job-01",
        "targetSelection": "SNAPSHOT",
        "status": "IN_PROGRESS",
        "targets": [
            "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"
        ],
        "description": "example job test",
        "presignedUrlConfig": {},
        "jobExecutionsRolloutConfig": {},
        "createdAt": 1560787022.733,
        "lastUpdatedAt": 1560787026.294,
        "jobProcessDetails": {
            "numberOfCanceledThings": 0,
            "numberOfSucceededThings": 0,
            "numberOfFailedThings": 0,
            "numberOfRejectedThings": 0,
            "numberOfQueuedThings": 1,
            "numberOfInProgressThings": 0,
            "numberOfRemovedThings": 0,
            "numberOfTimedOutThings": 0
        },
        "timeoutConfig": {}
    }
}
```

Pour plus d'informations, consultez Creating and Managing Jobs (CLI) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateJobà la section Référence des AWS CLI commandes.

`update-mitigation-action`

L'exemple de code suivant montre comment utiliser`update-mitigation-action`.

AWS CLI

Pour mettre à jour une mesure d'atténuation

L'`update-mitigation-action`exemple suivant met à jour l'action d'atténuation spécifiée nomméeAddThingsToQuarantineGroupAction, modifie le nom du groupe d'objets et la définit `overrideDynamicGroups` sur`false`. Vous pouvez vérifier vos modifications à l'aide de la `describe-mitigation-action` commande.

```
aws iot update-mitigation-action \
    --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",
 \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":
[\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"
```

Sortie :

```
{
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
    "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"
}
```

Pour plus d'informations, voir UpdateMitigationAction (Mitigation Action Commands) dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateMitigationActionà la section Référence des AWS CLI commandes.

`update-provisioning-template`

L'exemple de code suivant montre comment utiliser`update-provisioning-template`.

AWS CLI

Pour mettre à jour un modèle de provisionnement

L'`update-provisioning-template`exemple suivant modifie la description et le rôle arn du modèle de provisionnement spécifié et active le modèle.

```
aws iot update-provisioning-template \
    --template-name widget-template \
    --enabled \
    --description "An updated provisioning template for widgets" \
    --provisioning-role-arn arn:aws:iam::504350838278:role/Provision_role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section AWS IoT Secure Tunneling dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous UpdateProvisioningTemplateà la section Référence des AWS CLI commandes.

`update-role-alias`

L'exemple de code suivant montre comment utiliser`update-role-alias`.

AWS CLI

Pour mettre à jour un alias de rôle

L'`update-role-alias`exemple suivant met à jour l'alias du `LightBulbRole` rôle.

```
aws iot update-role-alias \
    --role-alias LightBulbRole \
    --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Sortie :

```
{
    "roleAlias": "LightBulbRole",
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

Pour plus d'informations, consultez UpdateRoleAliasle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateRoleAliasà la section Référence des AWS CLI commandes.

`update-scheduled-audit`

L'exemple de code suivant montre comment utiliser`update-scheduled-audit`.

AWS CLI

Pour mettre à jour une définition d'audit planifié

L'`update-scheduled-audit`exemple suivant modifie les noms des contrôles cibles pour un audit planifié d' AWS IoT Device Defender.

```
aws iot update-scheduled-audit \
    --scheduled-audit-name WednesdayCertCheck \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Sortie :

```
{
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

Pour plus d'informations, consultez la section Commandes d'audit du guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateScheduledAudità la section Référence des AWS CLI commandes.

`update-security-profile`

L'exemple de code suivant montre comment utiliser`update-security-profile`.

AWS CLI

Pour modifier un profil de sécurité

L'`update-security-profile`exemple suivant met à jour à la fois la description et les comportements d'un profil de sécurité AWS IoT Device Defender.

```
aws iot update-security-profile \
    --security-profile-name PossibleIssue \
    --security-profile-description "Check to see if authorization fails 12 times in
 5 minutes or if cellular bandwidth exceeds 128"  \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":12},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

Sortie :

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
    "securityProfileDescription": "check to see if authorization fails 12 times in 5
 minutes or if cellular bandwidth exceeds 128",
    "behaviors": [
        {
            "name": "CellularBandwidth",
            "metric": "aws:message-byte-size",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 128
                },
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        {
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "less-than",
                "value": {
                    "count": 12
                },
```

```
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        }
    ],
    "version": 2,
    "creationDate": 1560278102.528,
    "lastModifiedDate": 1560352711.207
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateSecurityProfileà la section Référence des AWS CLI commandes.

`update-stream`

L'exemple de code suivant montre comment utiliser`update-stream`.

AWS CLI

Pour mettre à jour un stream

L'`update-stream`exemple suivant met à jour un flux existant. La version du stream est incrémentée d'une unité.

```
aws iot update-stream \
    --cli-input-json file://update-stream.json
```

Contenu de `update-stream.json` :

```
{
    "streamId": "stream12345",
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
    "files": [
        {
            "fileId": 123,
            "s3Location": {
                "bucket":"codesign-ota-bucket",
                "key":"48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
```

```
            }
        }
    ]
    "roleArn": "arn:aws:iam:us-west-2:123456789012:role/service-role/
my_ota_stream_role"
}
```

Sortie :

```
{
    "streamId": "stream12345",
    "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
    "streamVersion": 2
}
```

Pour plus d'informations, consultez UpdateStreamle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateStreamà la section Référence des AWS CLI commandes.

`update-thing-group`

L'exemple de code suivant montre comment utiliser`update-thing-group`.

AWS CLI

Pour mettre à jour la définition d'un groupe d'objets

L'`update-thing-group`exemple suivant met à jour la définition du groupe d'objets spécifié, en modifiant la description et deux attributs.

```
aws iot update-thing-group \
    --thing-group-name HalogenBulbs \
    --thing-group-properties "thingGroupDescription=\"Halogen bulb group\",
 attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

Sortie :

```
{
    "version": 2
```

```
}
```

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateThingGroupà la section Référence des AWS CLI commandes.

`update-thing-groups-for-thing`

L'exemple de code suivant montre comment utiliser`update-thing-groups-for-thing`.

AWS CLI

Pour modifier les groupes auxquels appartient un objet

L'`update-thing-groups-for-thing`exemple suivant supprime l'objet nommé `MyLightBulb` du groupe nommé `DeadBulbs` et l'ajoute `replaceableItems` au groupe nommé en même temps.

```
aws iot update-thing-groups-for-thing \
    --thing-name MyLightBulb \
    --thing-groups-to-add "replaceableItems" \
    --thing-groups-to-remove "DeadBulbs"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Thing Groups dans le guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateThingGroupsForThingà la section Référence des AWS CLI commandes.

`update-thing`

L'exemple de code suivant montre comment utiliser`update-thing`.

AWS CLI

Pour associer un objet à un type d'objet

L'`update-thing`exemple suivant associe un objet du registre AWS IoT à un type d'objet. Lorsque vous établissez l'association, vous fournissez des valeurs pour les attributs définis par le type d'objet.

```
aws iot update-thing \
    --thing-name "MyOtherLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

Cette commande ne produit aucune sortie. Utilisez la `describe-thing` commande pour voir le résultat.

Pour plus d'informations, consultez la section Types d'objets dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateThingà la section Référence des AWS CLI commandes.

`update-topic-rule-destination`

L'exemple de code suivant montre comment utiliser`update-topic-rule-destination`.

AWS CLI

Exemple 1 : pour activer une destination de règle de sujet

L'`update-topic-rule-destination`exemple suivant active le trafic vers une destination de règle de rubrique.

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status ENABLED
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation d'une destination de règle de sujet dans le Guide du développeur de l'AWS IoT.

Exemple 2 : pour désactiver la destination d'une règle de sujet

L'`update-topic-rule-destination`exemple suivant désactive le trafic vers une destination de règle de rubrique.

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status DISABLED
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation d'une destination de règle de sujet dans le Guide du développeur de l'AWS IoT.

Exemple 3 : pour envoyer un nouveau message de confirmation

L'`update-topic-rule-destination`exemple suivant envoie un nouveau message de confirmation pour une destination de règle de rubrique.

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status IN_PROGRESS
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Envoi d'un nouveau message de confirmation dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateTopicRuleDestinationà la section Référence des AWS CLI commandes.

`validate-security-profile-behaviors`

L'exemple de code suivant montre comment utiliser`validate-security-profile-behaviors`.

AWS CLI

Exemple 1 : pour valider les paramètres de comportement d'un profil de sécurité

L'`validate-security-profile-behaviors`exemple suivant valide un ensemble de comportements bien formé et correct pour un profil de sécurité AWS IoT Device Defender.

```
aws iot validate-security-profile-behaviors \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":12},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

Sortie :

```
{
    "valid": true,
    "validationErrors": []
}
```

Exemple 2 : pour valider les paramètres de comportement incorrects pour un profil de sécurité

L'`validate-security-profile-behaviors`exemple suivant valide un ensemble de comportements contenant une erreur pour un profil de sécurité AWS IoT Device Defender.

```
aws iot validate-security-profile-behaviors \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":12},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":100000,\"consecutiveDatapointsToClear
\":1}}]"
```

Sortie :

```
{
    "valid": false,
    "validationErrors": [
        {
            "errorMessage": "Behavior Authorization is malformed.
 consecutiveDatapointsToAlarm 100000 should be in range[1,10]"
        }
    ]
}
```

Pour plus d'informations, consultez la section Detect Commands dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ValidateSecurityProfileBehaviorsà la section Référence des AWS CLI commandes.

AWS IoT Analytics exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Analytics.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-put-message`

L'exemple de code suivant montre comment utiliser`batch-put-message`.

AWS CLI

Pour envoyer un message à une chaîne

L'`batch-put-message`exemple suivant envoie un message au canal spécifié.

```
aws iotanalytics batch-put-message \
    --cli-binary-format raw-in-base64-out \
    --cli-input-json file://batch-put-message.json
```

Contenu de `batch-put-message.json` :

```
{
```

```
    "channelName": "mychannel",
    "messages": [
        {
            "messageId": "0001",
            "payload": "eyAidGVtcGVyYXR1cmUiOiAyMCB9"
        }
    ]
}
```

Sortie :

```
{
    "batchPutMessageErrorEntries": []
}
```

Pour plus d'informations, consultez BatchPutMessagela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous BatchPutMessageà la section Référence des AWS CLI commandes.

`cancel-pipeline-reprocessing`

L'exemple de code suivant montre comment utiliser`cancel-pipeline-reprocessing`.

AWS CLI

Pour annuler le retraitement des données via un pipeline

L'`cancel-pipeline-reprocessing`exemple suivant annule le retraitement des données via le pipeline spécifié.

```
aws iotanalytics cancel-pipeline-reprocessing \
    --pipeline-name mypipeline \
    --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez CancelPipelineReprocessingla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous CancelPipelineReprocessingà la section Référence des AWS CLI commandes.

`create-channel`

L'exemple de code suivant montre comment utiliser`create-channel`.

AWS CLI

Pour créer une chaîne

L'`create-channel`exemple suivant crée un canal avec la configuration spécifiée. Un canal collecte des données à partir d'une rubrique MQTT et archive les messages bruts non traités avant de publier les données dans un pipeline.

```
aws iotanalytics create-channel \
    --cli-input-json file://create-channel.json
```

Contenu de `create-channel.json` :

```
{
    "channelName": "mychannel",
    "retentionPeriod": {
        "unlimited": true
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

Sortie :

```
{
    "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",
    "channelName": "mychannel",
    "retentionPeriod": {
        "unlimited": true
    }
}
```

Pour plus d'informations, consultez CreateChannella référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous CreateChannelà la section Référence des AWS CLI commandes.

`create-dataset-content`

L'exemple de code suivant montre comment utiliser`create-dataset-content`.

AWS CLI

Pour créer le contenu d'un ensemble de données

L'`create-dataset-content`exemple suivant crée le contenu de l'ensemble de données spécifié en appliquant a `queryAction` (requête SQL) ou a `containerAction` (exécution d'une application conteneurisée).

```
aws iotanalytics create-dataset-content \
    --dataset-name mydataset
```

Sortie :

```
{
    "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"
}
```

Pour plus d'informations, consultez CreateDatasetContentla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous CreateDatasetContentà la section Référence des AWS CLI commandes.

`create-dataset`

L'exemple de code suivant montre comment utiliser`create-dataset`.

AWS CLI

Pour créer un jeu de données

L'`create-dataset`exemple suivant crée un ensemble de données. Un ensemble de données stocke les données extraites d'un magasin de données en appliquant une `queryAction` (requête SQL) ou a `containerAction` (exécution d'une application conteneurisée). Cette opération crée

le squelette d'un ensemble de données. Vous pouvez remplir le jeu de données manuellement en appelant `CreateDatasetContent` ou automatiquement selon un paramètre `trigger` que vous spécifiez.

```
aws iotanalytics create-dataset \
    --cli-input-json file://create-dataset.json
```

Contenu de `create-dataset.json` :

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM mydatastore"
            }
        }
    ],
    "retentionPeriod": {
        "unlimited": true
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

Sortie :

```
{
    "datasetName": "mydataset",
    "retentionPeriod": {
        "unlimited": true
    },
    "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

Pour plus d'informations, consultez CreateDatasetla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous CreateDatasetà la section Référence des AWS CLI commandes.

`create-datastore`

L'exemple de code suivant montre comment utiliser`create-datastore`.

AWS CLI

Pour créer un magasin de données

L'`create-datastore`exemple suivant crée un magasin de données, qui est un référentiel pour les messages.

```
aws iotanalytics create-datastore \
    --cli-input-json file://create-datastore.json
```

Contenu de `create-datastore.json` :

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 90
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

Sortie :

```
{
    "datastoreName": "mydatastore",
    "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
mydatastore",
    "retentionPeriod": {
        "numberOfDays": 90,
        "unlimited": false
```

```
    }
}
```

Pour plus d'informations, consultez CreateDatastorela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous CreateDatastoreà la section Référence des AWS CLI commandes.

`create-pipeline`

L'exemple de code suivant montre comment utiliser`create-pipeline`.

AWS CLI

Créez un pipeline d'analyse IoT

L'`create-pipeline`exemple suivant crée un pipeline. Un pipeline consomme les messages à partir d'un canal et vous permet de traiter les messages avant de les stocker dans un magasin de données. Vous devez spécifier à la fois une activité de canal et une activité de stockage de données et, éventuellement, jusqu'à 23 activités supplémentaires dans le `pipelineActivities` tableau.

```
aws iotanalytics create-pipeline \
    --cli-input-json file://create-pipeline.json
```

Contenu de `create-pipeline.json` :

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "myChannelActivity",
                "channelName": "mychannel",
                "next": "myMathActivity"
            }
        },
        {
            "datastore": {
                "name": "myDatastoreActivity",
                "datastoreName": "mydatastore"
            }
```

```
        },
        {
            "math": {
                "name": "myMathActivity",
                "math": "((temp - 32) * 5.0) / 9.0",
                "attribute": "tempC",
                "next": "myDatastoreActivity"
            }
        }
    ],
    "tags": [
        {
            "key": "Environment",
            "value": "Beta"
        }
    ]
}
```

Sortie :

```
{
    "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
    "pipelineName": "mypipeline"
}
```

Pour plus d'informations, consultez CreatePipelinela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous CreatePipelineà la section Référence des AWS CLI commandes.

`delete-channel`

L'exemple de code suivant montre comment utiliser`delete-channel`.

AWS CLI

Supprimer un canal IoT Analytics

L'`delete-channel`exemple suivant supprime le canal spécifié.

```
aws iotanalytics delete-channel \
```

```
    --channel-name mychannel
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteChannella référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DeleteChannelà la section Référence des AWS CLI commandes.

`delete-dataset-content`

L'exemple de code suivant montre comment utiliser`delete-dataset-content`.

AWS CLI

Pour supprimer le contenu d'un ensemble de données

L'`delete-dataset-content`exemple suivant supprime le contenu de l'ensemble de données spécifié.

```
aws iotanalytics delete-dataset-content \
    --dataset-name mydataset
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteDatasetContentla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DeleteDatasetContentà la section Référence des AWS CLI commandes.

`delete-dataset`

L'exemple de code suivant montre comment utiliser`delete-dataset`.

AWS CLI

Pour supprimer un ensemble de données

L'`delete-dataset`exemple suivant supprime l'ensemble de données spécifié. Il n'est pas nécessaire de supprimer le contenu de l'ensemble de données avant d'effectuer cette opération.

```
aws iotanalytics delete-dataset \
```

```
    --dataset-name mydataset
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteDatasetla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DeleteDatasetà la section Référence des AWS CLI commandes.

`delete-datastore`

L'exemple de code suivant montre comment utiliser`delete-datastore`.

AWS CLI

Pour supprimer un magasin de données

L'`delete-datastore`exemple suivant supprime le magasin de données spécifié.

```
aws iotanalytics delete-datastore \
    --datastore-name mydatastore
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteDatastorela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DeleteDatastoreà la section Référence des AWS CLI commandes.

`delete-pipeline`

L'exemple de code suivant montre comment utiliser`delete-pipeline`.

AWS CLI

Pour supprimer un pipeline

L'`delete-pipeline`exemple suivant supprime le pipeline spécifié.

```
aws iotanalytics delete-pipeline \
    --pipeline-name mypipeline
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeletePipelinela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DeletePipelineà la section Référence des AWS CLI commandes.

`describe-channel`

L'exemple de code suivant montre comment utiliser`describe-channel`.

AWS CLI

Pour récupérer des informations sur une chaîne

L'`describe-channel`exemple suivant affiche les détails, y compris les statistiques, pour le canal spécifié.

```
aws iotanalytics describe-channel \
    --channel-name mychannel \
    --include-statistics
```

Sortie :

```
{
    "statistics": {
        "size": {
            "estimatedSizeInBytes": 402.0,
            "estimatedOn": 1561504380.0
        }
    },
    "channel": {
        "status": "ACTIVE",
        "name": "mychannel",
        "lastUpdateTime": 1557860351.001,
        "creationTime": 1557860351.001,
        "retentionPeriod": {
            "unlimited": true
        },
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
    }
}
```

Pour plus d'informations, consultez DescribeChannella référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DescribeChannelà la section Référence des AWS CLI commandes.

`describe-dataset`

L'exemple de code suivant montre comment utiliser`describe-dataset`.

AWS CLI

Pour récupérer des informations sur un ensemble de données

L'`describe-dataset`exemple suivant affiche les détails de l'ensemble de données spécifié.

```
aws iotanalytics describe-dataset \
    --dataset-name mydataset
```

Sortie :

```
{
    "dataset": {
        "status": "ACTIVE",
        "contentDeliveryRules": [],
        "name": "mydataset",
        "lastUpdateTime": 1557859240.658,
        "triggers": [],
        "creationTime": 1557859240.658,
        "actions": [
            {
                "actionName": "query_32",
                "queryAction": {
                    "sqlQuery": "SELECT * FROM mydatastore",
                    "filters": []
                }
            }
        ],
        "retentionPeriod": {
            "numberOfDays": 90,
            "unlimited": false
        },
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
    }
```

```
}
```

Pour plus d'informations, consultez DescribeDatasetla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DescribeDatasetà la section Référence des AWS CLI commandes.

`describe-datastore`

L'exemple de code suivant montre comment utiliser`describe-datastore`.

AWS CLI

Pour récupérer des informations sur un magasin de données

L'`describe-datastore`exemple suivant affiche les détails, y compris les statistiques, du magasin de données spécifié.

```
aws iotanalytics describe-datastore \
    --datastore-name mydatastore \
    --include-statistics
```

Sortie :

```
{
    "datastore": {
        "status": "ACTIVE",
        "name": "mydatastore",
        "lastUpdateTime": 1557858971.02,
        "creationTime": 1557858971.02,
        "retentionPeriod": {
            "unlimited": true
        },
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/mydatastore"
    },
    "statistics": {
        "size": {
            "estimatedSizeInBytes": 397.0,
            "estimatedOn": 1561592040.0
        }
    }
}
```

Pour plus d'informations, consultez DescribeDatastorela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DescribeDatastoreà la section Référence des AWS CLI commandes.

`describe-logging-options`

L'exemple de code suivant montre comment utiliser`describe-logging-options`.

AWS CLI

Pour récupérer les options de journalisation actuelles

L'`describe-logging-options`exemple suivant affiche les options de journalisation actuelles d' AWS IoT Analytics.

```
aws iotanalytics describe-logging-options
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
        "enabled": true,
        "level": "ERROR"
    }
}
```

Pour plus d'informations, consultez DescribeLoggingOptionsla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DescribeLoggingOptionsà la section Référence des AWS CLI commandes.

`describe-pipeline`

L'exemple de code suivant montre comment utiliser`describe-pipeline`.

AWS CLI

Pour récupérer des informations sur un pipeline

L'`describe-pipeline`exemple suivant affiche les détails du pipeline spécifié.

```
aws iotanalytics describe-pipeline \
    --pipeline-name mypipeline
```

Sortie :

```
{
    "pipeline": {
        "activities": [
            {
                "channel": {
                    "channelName": "mychannel",
                    "name": "mychannel_28",
                    "next": "mydatastore_29"
                }
            },
            {
                "datastore": {
                    "datastoreName": "mydatastore",
                    "name": "mydatastore_29"
                }
            }
        ],
        "name": "mypipeline",
        "lastUpdateTime": 1561676362.515,
        "creationTime": 1557859124.432,
        "reprocessingSummaries": [
            {
                "status": "SUCCEEDED",
                "creationTime": 1561676362.189,
                "id": "6ad2764f-fb13-4de3-b101-4e74af03b043"
            }
        ],
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/mypipeline"
    }
}
```

Pour plus d'informations, consultez DescribePipelinela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous DescribePipelineà la section Référence des AWS CLI commandes.

`get-dataset-content`

L'exemple de code suivant montre comment utiliser`get-dataset-content`.

AWS CLI

Pour récupérer le contenu d'un ensemble de données

L'`get-dataset-content`exemple suivant récupère le contenu d'un ensemble de données tel que URIs présigné.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Sortie :

```
{
    "status": {
        "state": "SUCCEEDED"
    },
    "timestamp": 1557863215.995,
    "entries": [
        {
            "dataURI": "https://aws-radiant-
dataset-12345678-1234-1234-1234-123456789012.s3.us-west-2.amazonaws.com/
results/12345678-e8b3-46ba-b2dd-efe8d86cf385.csv?X-Amz-Security-Token=...-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190628T173437Z&X-Amz-SignedHeaders=host&X-
Amz-Expires=7200&X-Amz-Credential=...F20190628%2Fus-west-2%2Fs3%2Faws4_request&X-
Amz-Signature=..."
        }
    ]
}
```

Pour plus d'informations, consultez GetDatasetContentle guide.

- Pour plus de détails sur l'API, reportez-vous GetDatasetContentà la section Référence des AWS CLI commandes.

`list-channels`

L'exemple de code suivant montre comment utiliser`list-channels`.

AWS CLI

Pour récupérer une liste de chaînes

L'`list-channels`exemple suivant affiche des informations récapitulatives sur les chaînes disponibles.

```
aws iotanalytics list-channels
```

Sortie :

```
{
    "channelSummaries": [
        {
            "status": "ACTIVE",
            "channelName": "mychannel",
            "creationTime": 1557860351.001,
            "lastUpdateTime": 1557860351.001
        }
    ]
}
```

Pour plus d'informations, consultez ListChannelsla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous ListChannelsà la section Référence des AWS CLI commandes.

`list-dataset-contents`

L'exemple de code suivant montre comment utiliser`list-dataset-contents`.

AWS CLI

Pour répertorier les informations relatives au contenu du jeu de données

L'`list-dataset-contents`exemple suivant répertorie les informations relatives au contenu des ensembles de données qui ont été créés.

```
aws iotanalytics list-dataset-contents \
    --dataset-name mydataset
```

Sortie :

```
{
    "datasetContentSummaries": [
        {
            "status": {
                "state": "SUCCEEDED"
            },
            "scheduleTime": 1557863215.995,
            "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",
            "creationTime": 1557863215.995
        }
    ]
}
```

Pour plus d'informations, consultez ListDatasetContentsla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous ListDatasetContentsà la section Référence des AWS CLI commandes.

`list-datasets`

L'exemple de code suivant montre comment utiliser`list-datasets`.

AWS CLI

Pour récupérer des informations sur les ensembles de données

L'`list-datasets`exemple suivant répertorie des informations récapitulatives sur les ensembles de données disponibles.

```
aws iotanalytics list-datasets
```

Sortie :

```
{
    "datasetSummaries": [
        {
            "status": "ACTIVE",
            "datasetName": "mydataset",
            "lastUpdateTime": 1557859240.658,
            "triggers": [],
```

```
            "creationTime": 1557859240.658,
            "actions": [
                {
                    "actionName": "query_32",
                    "actionType": "QUERY"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez ListDatasetsla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous ListDatasetsà la section Référence des AWS CLI commandes.

`list-datastores`

L'exemple de code suivant montre comment utiliser`list-datastores`.

AWS CLI

Pour récupérer une liste de magasins de données

L'`list-datastores`exemple suivant affiche des informations récapitulatives sur les magasins de données disponibles.

```
aws iotanalytics list-datastores
```

Sortie :

```
{
    "datastoreSummaries": [
        {
            "status": "ACTIVE",
            "datastoreName": "mydatastore",
            "creationTime": 1557858971.02,
            "lastUpdateTime": 1557858971.02
        }
    ]
}
```

Pour plus d'informations, consultez ListDatastoresla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous ListDatastoresà la section Référence des AWS CLI commandes.

`list-pipelines`

L'exemple de code suivant montre comment utiliser`list-pipelines`.

AWS CLI

Pour récupérer une liste de pipelines

L'`list-pipelines`exemple suivant affiche la liste des pipelines disponibles.

```
aws iotanalytics list-pipelines
```

Sortie :

```
{
    "pipelineSummaries": [
        {
            "pipelineName": "mypipeline",
            "creationTime": 1557859124.432,
            "lastUpdateTime": 1557859124.432,
            "reprocessingSummaries": []
        }
    ]
}
```

Pour plus d'informations, consultez ListPipelinesla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous ListPipelinesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises que vous avez attachées à la ressource spécifiée.

```
aws iotanalytics list-tags-for-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

Sortie :

```
{
    "tags": [
        {
            "value": "bar",
            "key": "foo"
        }
    ]
}
```

Pour plus d'informations, consultez ListTagsForResourcela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-logging-options`

L'exemple de code suivant montre comment utiliser`put-logging-options`.

AWS CLI

Pour définir ou mettre à jour les options de journalisation

L'`put-logging-options`exemple suivant définit ou met à jour les options de journalisation d' AWS IoT Analytics. Si vous mettez à jour la valeur d'un `loggingOptions` champ, la modification peut prendre jusqu'à une minute pour prendre effet. De même, si vous modifiez la politique associée au rôle que vous avez spécifié dans le champ « ROLearn » (par exemple, pour corriger une politique non valide), la prise en compte de cette modification peut prendre jusqu'à cinq minutes.

```
aws iotanalytics put-logging-options \
    --cli-input-json file://put-logging-options.json
```

Contenu de `put-logging-options.json` :

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
        "level": "ERROR",
        "enabled": true
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez PutLoggingOptionsla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous PutLoggingOptionsà la section Référence des AWS CLI commandes.

`run-pipeline-activity`

L'exemple de code suivant montre comment utiliser`run-pipeline-activity`.

AWS CLI

Pour simuler une activité de pipeline

L'`run-pipeline-activity`exemple suivant simule les résultats de l'exécution d'une activité de pipeline sur la charge utile d'un message.

```
aws iotanalytics run-pipeline-activity \
    --pipeline-activity file://maths.json \
    --payloads file://payloads.json
```

Contenu de `maths.json` :

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

Contenu de `payloads.json` :

```
[
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
]
```

Sortie :

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

Pour plus d'informations, consultez RunPipelineActivityla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous RunPipelineActivityà la section Référence des AWS CLI commandes.

`sample-channel-data`

L'exemple de code suivant montre comment utiliser`sample-channel-data`.

AWS CLI

Pour récupérer des exemples de messages à partir d'un canal

L'`sample-channel-data`exemple suivant extrait un échantillon de messages provenant du canal spécifié ingérés pendant la période spécifiée. Vous pouvez récupérer jusqu'à 10 messages.

```
aws iotanalytics sample-channel-data \
    --channel-name mychannel
```

Sortie :

```
{
    "payloads": [
        "eyAidGVtcGVyYXR1cmUiOiAyMCB9",
        "eyAiZm9vIjogImJhciIgfQ=="
    ]
```

```
}
```

Pour plus d'informations, consultez SampleChannelDatala référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous SampleChannelDataà la section Référence des AWS CLI commandes.

`start-pipeline-reprocessing`

L'exemple de code suivant montre comment utiliser`start-pipeline-reprocessing`.

AWS CLI

Pour démarrer le retraitement du pipeline

L'`start-pipeline-reprocessing`exemple suivant lance le retraitement des données de message brutes via le pipeline spécifié.

```
aws iotanalytics start-pipeline-reprocessing \
    --pipeline-name mypipeline
```

Sortie :

```
{
    "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"
}
```

Pour plus d'informations, consultez StartPipelineReprocessingla référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous StartPipelineReprocessingà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter ou modifier des balises pour une ressource

L'`tag-resource`exemple suivant ajoute ou modifie les balises associées à la ressource spécifiée.

```
aws iotanalytics tag-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \
    --tags "[{\"key\": \"Environment\", \"value\": \"Production\"}]"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez TagResourcela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`untag-resource`exemple suivant supprime les balises portant les noms de clé spécifiés de la ressource spécifiée.

```
aws iotanalytics untag-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \
    --tag-keys "[\"Environment\"]"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UntagResource < https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API _ UntagResource .html > dans le manuel AWS IoT Analytics API Reference.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-channel`

L'exemple de code suivant montre comment utiliser`update-channel`.

AWS CLI

Pour modifier une chaîne

L'`update-channel`exemple suivant modifie les paramètres du canal spécifié.

```
aws iotanalytics update-channel \
    --cli-input-json file://update-channel.json
```

Contenu de `update-channel.json` :

```
{
    "channelName": "mychannel",
    "retentionPeriod": {
        "numberOfDays": 92
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UpdateChannella référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous UpdateChannelà la section Référence des AWS CLI commandes.

`update-dataset`

L'exemple de code suivant montre comment utiliser`update-dataset`.

AWS CLI

Pour mettre à jour un ensemble de données

L'`update-dataset`exemple suivant modifie les paramètres de l'ensemble de données spécifié.

```
aws iotanalytics update-dataset \
    --cli-input-json file://update-dataset.json
```

Contenu de `update-dataset.json` :

```
{
```

```
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName": "myDatasetUpdateAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM mydatastore"
            }
        }
    ],
    "retentionPeriod": {
        "numberOfDays": 92
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UpdateDataset < https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API _ UpdateDataset .html > dans le manuel AWS IoT Analytics API Reference.

- Pour plus de détails sur l'API, reportez-vous UpdateDatasetà la section Référence des AWS CLI commandes.

`update-datastore`

L'exemple de code suivant montre comment utiliser`update-datastore`.

AWS CLI

Pour mettre à jour un magasin de données

L'`update-datastore`exemple suivant modifie les paramètres du magasin de données spécifié.

```
aws iotanalytics update-datastore \
    --cli-input-json file://update-datastore.json
```

Contenu du fichier update-datastore.json :

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 93
```

```
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UpdateDatastorela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous UpdateDatastoreà la section Référence des AWS CLI commandes.

`update-pipeline`

L'exemple de code suivant montre comment utiliser`update-pipeline`.

AWS CLI

Pour mettre à jour un pipeline

L'`update-pipeline`exemple suivant modifie les paramètres du pipeline spécifié. Vous devez spécifier à la fois une activité de canal et une activité de stockage de données et, éventuellement, jusqu'à 23 activités supplémentaires, dans le `pipelineActivities` tableau.

```
aws iotanalytics update-pipeline \
    --cli-input-json file://update-pipeline.json
```

Contenu de update-pipeline.json :

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "myChannelActivity",
                "channelName": "mychannel",
                "next": "myMathActivity"
            }
        },
        {
            "datastore": {
                "name": "myDatastoreActivity",
                "datastoreName": "mydatastore"
```

```
            }
        },
        {
            "math": {
                "name": "myMathActivity",
                "math": "(((temp - 32) * 5.0) / 9.0) + 273.15",
                "attribute": "tempK",
                "next": "myDatastoreActivity"
            }
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UpdatePipelinela référence des API AWS IoT Analytics.

- Pour plus de détails sur l'API, reportez-vous UpdatePipelineà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Device Advisor AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Device Advisor.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-suite-definition`

L'exemple de code suivant montre comment utiliser`create-suite-definition`.

AWS CLI

Exemple 1 : pour créer une suite de tests IoT Device Advisor

L'`create-suite-definition`exemple suivant crée une suite de tests Device Advisor dans l' AWS IoT avec la configuration de définition de suite spécifiée.

```
aws iotdeviceadvisor create-suite-definition \
    --suite-definition-configuration '{ \
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": false, \
        "rootGroup": "{\"configuration\":{},\"tests\":[{\"name\":\"MQTT Connect\",
\"configuration\":{\"EXECUTION_TIMEOUT\":120},\"tests\":[{\"name\":\"MQTT_Connect\",
\"configuration\":{},\"test\":{\"id\":\"MQTT_Connect\",\"testCase\":null,\"version
\":\"0.0.0\"}}]}]}", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

Sortie :

```
{
    "suiteDefinitionId": "0jtsgio7yenu",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/0jtsgio7yenu",
    "suiteDefinitionName": "TestSuiteName",
    "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

Pour plus d'informations, consultez la section Création d'une définition de suite de tests dans le guide du développeur AWS IoT Core.

Exemple 2 : pour créer une suite de tests de qualification IoT Device Advisor Latest

L'`create-suite-definition`exemple suivant crée une suite de tests de qualification Device Advisor avec la dernière version de l' AWS IoT avec la configuration de définition de suite spécifiée.

```
aws iotdeviceadvisor create-suite-definition \
    --suite-definition-configuration '{ \
        "suiteDefinitionName": "TestSuiteName", \
```

```
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": true, \
        "rootGroup": "", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

Sortie :

```
{
    "suiteDefinitionId": "txgsuolk2myj",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/txgsuolk2myj",
    "suiteDefinitionName": "TestSuiteName",
    "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

Pour plus d'informations, consultez la section Création d'une définition de suite de tests dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous CreateSuiteDefinitionà la section Référence des AWS CLI commandes.

`delete-suite-definition`

L'exemple de code suivant montre comment utiliser`delete-suite-definition`.

AWS CLI

Pour supprimer la suite de tests IoT Device Advisor

L'`delete-suite-definition`exemple suivant supprime la suite de tests Device Advisor avec l'ID de définition de suite spécifié.

```
aws iotdeviceadvisor delete-suite-definition \
    --suite-definition-id 0jtsgio7yenu
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteSuiteDefinitionle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteSuiteDefinitionà la section Référence des AWS CLI commandes.

`get-endpoint`

L'exemple de code suivant montre comment utiliser`get-endpoint`.

AWS CLI

Exemple 1 : pour obtenir des informations sur un point de terminaison au niveau du compte IoT Device Advisor

L'`get-endpoint`exemple suivant permet d'obtenir les informations relatives à un point de terminaison de test au niveau du compte Device Advisor.

```
aws iotdeviceadvisor get-endpoint
```

Sortie :

```
{
    "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Exemple 2 : Pour obtenir des informations sur un point de terminaison au niveau de l'appareil IoT Device Advisor

L'`get-endpoint`exemple suivant obtient les informations relatives à un point de terminaison de test Device Advisor au niveau de l'appareil avec le thing-arn ou le certificate-arn spécifié.

```
aws iotdeviceadvisor get-endpoint \
    --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIotThing
```

Sortie :

```
{
    "endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Pour plus d'informations, consultez la section Obtenir un point de terminaison de test dans le guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous GetEndpointà la section Référence des AWS CLI commandes.

`get-suite-definition`

L'exemple de code suivant montre comment utiliser`get-suite-definition`.

AWS CLI

Pour obtenir des informations sur une suite de tests IoT Device Advisor

L'`get-suite-definition`exemple suivant permet d'obtenir les informations relatives à une suite de tests Advice Advisor avec l'ID de définition de suite spécifié.

```
aws iotdeviceadvisor get-suite-definition \
    --suite-definition-id qqcsmtyyjabl
```

Sortie :

```
{
    "suiteDefinitionId": "qqcsmtyyjabl",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/qqcsmtyyjabl",
    "suiteDefinitionVersion": "v1",
    "latestVersion": "v1",
    "suiteDefinitionConfiguration": {
        "suiteDefinitionName": "MQTT connection",
        "devices": [],
        "intendedForQualification": false,
        "isLongDurationTest": false,
        "rootGroup": "{\"configuration\":{},\"tests\":[{\"id\":\"uta5d9j1kvwc\",
\"name\":\"Test group 1\",\"configuration\":{},\"tests\":[{\"id\":\"awr8pq5vc9yp\",
\"name\":\"MQTT Connect\",\"configuration\":{},\"test\":{\"id\":\"MQTT_Connect\",
\"testCase\":null,\"version\":\"0.0.0\"}}]}]}",
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole",
        "protocol": "MqttV3_1_1"
    },
    "createdAt": "2022-11-11T22:28:52.389000-05:00",
    "lastModifiedAt": "2022-11-11T22:28:52.389000-05:00",
    "tags": {}
}
```

Pour plus d'informations, voir Obtenir une définition de suite de tests dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous GetSuiteDefinitionà la section Référence des AWS CLI commandes.

`get-suite-run-report`

L'exemple de code suivant montre comment utiliser`get-suite-run-report`.

AWS CLI

Pour obtenir des informations sur une suite de tests de qualification pour IoT Device Advisor, exécutez un rapport

L'`get-suite-run-report`exemple suivant permet d'obtenir le lien de téléchargement du rapport pour une suite de tests de qualification de Device Advisor exécutée avec succès avec l'ID de définition de suite et l'ID d'exécution de suite spécifiés.

```
aws iotdeviceadvisor get-suite-run-report \
    --suite-definition-id ztvb5aek4w4x \
    --suite-run-id p6awv83nre6v
```

Sortie :

```
{
    "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-
prod.s3.amazonaws.com/report.downloadlink"
}
```

Pour plus d'informations, voir Obtenir un rapport de qualification pour une suite de tests de qualification exécutée avec succès dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous GetSuiteRunReportà la section Référence des AWS CLI commandes.

`get-suite-run`

L'exemple de code suivant montre comment utiliser`get-suite-run`.

AWS CLI

Pour obtenir des informations sur l'état d'exécution d'une suite de tests IoT Device Advisor

L'`get-suite-run`exemple suivant permet d'obtenir les informations relatives à l'état d'exécution d'une suite de tests Device Advisor avec l'ID de définition de suite et l'ID d'exécution de la suite spécifiés.

```
aws iotdeviceadvisor get-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-run-id nzlfyhaa18oa
```

Sortie :

```
{
    "suiteDefinitionId": "qqcsmtyyjabl",
    "suiteDefinitionVersion": "v1",
    "suiteRunId": "nzlfyhaa18oa",
    "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/
qqcsmtyyjabl/nzlfyhaa18oa",
    "suiteRunConfiguration": {
        "primaryDevice": {
            "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing",
            "certificateArn": "arn:aws:iot:us-east-1:123456789012:cert/certFile"
        },
        "parallelRun": false
    },
    "testResult": {
        "groups": [
            {
                "groupId": "uta5d9j1kvwc",
                "groupName": "Test group 1",
                "tests": [
                    {
                        "testCaseRunId": "2ve2twrqyr0s",
                        "testCaseDefinitionId": "awr8pq5vc9yp",
                        "testCaseDefinitionName": "MQTT Connect",
                        "status": "PASS",
                        "startTime": "2022-11-12T00:01:53.693000-05:00",
                        "endTime": "2022-11-12T00:02:15.443000-05:00",
                        "logUrl": "https://console.aws.amazon.com/
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws/iot/deviceadvisor/
qqcsmtyyjabl;stream=nzlfyhaa18oa_2ve2twrqyr0s",
```

```
                        "warnings": "null",
                        "failure": "null"
                    }
                ]
            }
        ]
    },
    "startTime": "2022-11-12T00:01:52.673000-05:00",
    "endTime": "2022-11-12T00:02:16.496000-05:00",
    "status": "PASS",
    "tags": {}
}
```

Pour plus d'informations, voir Exécuter une suite de tests dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous GetSuiteRunà la section Référence des AWS CLI commandes.

`list-suite-definitions`

L'exemple de code suivant montre comment utiliser`list-suite-definitions`.

AWS CLI

Exemple 1 : pour répertorier les suites de tests IoT Device Advisor que vous avez créées

L'`list-suite-definitions`exemple suivant répertorie jusqu'à 25 suites de tests Device Advisor que vous avez créées dans AWS l'IoT. Si vous avez plus de 25 suites de tests, le « NextToken » sera affiché dans la sortie. Vous pouvez utiliser ce « NextToken » pour afficher le reste des suites de tests que vous avez créées.

```
aws iotdeviceadvisor list-suite-definitions
```

Sortie :

```
{
    "suiteDefinitionInformationList": [
        {
            "suiteDefinitionId": "3hsn88h4p2g5",
            "suiteDefinitionName": "TestSuite1",
            "defaultDevices": [
```

```
                {
                    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"
                }
            ],
            "intendedForQualification": false,
            "isLongDurationTest": false,
            "protocol": "MqttV3_1_1",
            "createdAt": "2022-11-17T14:15:56.830000-05:00"
        },
        {
            ......
        }
    ],
    "nextToken": "nextTokenValue"
}
```

Exemple 2 : pour répertorier les suites de test IoT Device Advisor que vous avez créées avec les paramètres spécifiés

L'`list-suite-definitions`exemple suivant répertorie les suites de tests Device Advisor que vous avez créées dans AWS l'IoT avec le nombre maximum de résultats spécifié. Si vous avez plus de suites de tests que le nombre maximum, le « NextToken » sera affiché dans la sortie. Si vous avez « NextToken », vous pouvez utiliser « NextToken » pour afficher les suites de tests que vous avez créées et qui n'étaient pas affichées auparavant.

```
aws iotdeviceadvisor list-suite-definitions \
    --max-result 1 \
    --next-token "nextTokenValue"
```

Sortie :

```
{
    "suiteDefinitionInformationList": [
        {
            "suiteDefinitionId": "ztvb5aew4w4x",
            "suiteDefinitionName": "TestSuite2",
            "defaultDevices": [],
            "intendedForQualification": true,
            "isLongDurationTest": false,
            "protocol": "MqttV3_1_1",
            "createdAt": "2022-11-17T14:15:56.830000-05:00"
```

```
        }
    ],
    "nextToken": "nextTokenValue"
}
```

Pour plus d'informations, consultez ListSuiteDefinitionsle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListSuiteDefinitionsà la section Référence des AWS CLI commandes.

`list-suite-runs`

L'exemple de code suivant montre comment utiliser`list-suite-runs`.

AWS CLI

Exemple 1 : pour répertorier toutes les informations relatives à l'IoT Device Advisor, la suite de tests indique l'état des exécutions

L'`list-suite-runs`exemple suivant répertorie toutes les informations relatives à l'état d'exécution d'une suite de tests Device Advisor avec l'ID de définition de suite spécifié. Si vous avez exécuté plus de 25 suites de tests, le « NextToken » sera affiché dans la sortie. Vous pouvez utiliser ce « NextToken » pour montrer le reste de la suite de tests exécutée.

```
aws iotdeviceadvisor list-suite-runs \
    --suite-definition-id ztvb5aew4w4x
```

Sortie :

```
{
    "suiteRunsList": [
        {
            "suiteDefinitionId": "ztvb5aew4w4x",
            "suiteDefinitionVersion": "v1",
            "suiteDefinitionName": "TestSuite",
            "suiteRunId": "p6awv89nre6v",
            "createdAt": "2022-12-01T16:33:14.212000-05:00",
            "startedAt": "2022-12-01T16:33:15.710000-05:00",
            "endAt": "2022-12-01T16:42:03.323000-05:00",
            "status": "PASS",
            "passed": 6,
```

```
            "failed": 0
        }
    ]
}
```

Exemple 2 : Pour répertorier les informations relatives à l'IoT, Device Advisor exécute l'état avec les paramètres spécifiés

L'`list-suite-runs`exemple suivant répertorie les informations relatives à l'état d'exécution d'une suite de tests Device Advisor avec l'ID de définition de suite spécifié et le numéro de résultat maximal spécifié. Si vous avez plus d'exécutions de suites de tests que le nombre maximum, le « NextToken » sera affiché dans la sortie. Si vous avez « NextToken », vous pouvez utiliser « NextToken » pour afficher les exécutions de la suite de tests qui n'étaient pas affichées auparavant.

```
aws iotdeviceadvisor list-suite-runs \
    --suite-definition-id qqcsmtyyjaml \
    --max-result 1 \
    --next-token "nextTokenValue"
```

Sortie :

```
{
    "suiteRunsList": [
        {
            "suiteDefinitionId": "qqcsmtyyjaml",
            "suiteDefinitionVersion": "v1",
            "suiteDefinitionName": "MQTT connection",
            "suiteRunId": "gz9vm2s6d2jy",
            "createdAt": "2022-12-01T20:10:27.079000-05:00",
            "startedAt": "2022-12-01T20:10:28.003000-05:00",
            "endAt": "2022-12-01T20:10:45.084000-05:00",
            "status": "STOPPED",
            "passed": 0,
            "failed": 0
        }
    ],
    "nextToken": "nextTokenValue"
}
```

Pour plus d'informations, consultez ListSuiteRunsle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListSuiteRunsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises associées à une ressource IoT Device Advisor

L'`list-tags-for-resource`exemple suivant répertorie les balises associées à une ressource Device Advisor. La ressource Device Advisor peut être un Suitedefinition-Arn ou un Suiterun-Arn.

```
aws iotdeviceadvisor list-tags-for-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny
```

Sortie :

```
{
    "tags": {
        "TestTagKey": "TestTagValue"
    }
}
```

Pour plus d'informations, consultez ListTagsForResourcela référence des API AWS IoT et les types de ressources définis par AWS IoT Core Device Advisor dans la référence d'autorisation de service.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-suite-run`

L'exemple de code suivant montre comment utiliser`start-suite-run`.

AWS CLI

Pour démarrer une suite de tests IoT Device Advisor, exécutez

L'`start-suite-run`exemple suivant répertorie les widgets disponibles dans votre AWS compte.

```
aws iotdeviceadvisor start-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-definition-version v1 \
    --suite-run-configuration '{"primaryDevice":{"thingArn": "arn:aws:iot:us-
east-1:123456789012:thing/MyIotThing","certificateArn":"arn:aws:iot:us-
east-1:123456789012:cert/certFile"}}'
```

Sortie :

```
{
    "suiteRunId": "pwmucgw7lt9s",
    "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/
qqcsmtyyjabl/pwmucgw7lk9s",
    "createdAt": "2022-12-02T15:43:05.581000-05:00"
}
```

Pour plus d'informations, voir Démarrer l'exécution d'une suite de tests dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous StartSuiteRunà la section Référence des AWS CLI commandes.

`stop-suite-run`

L'exemple de code suivant montre comment utiliser`stop-suite-run`.

AWS CLI

Pour arrêter une suite de tests IoT Device Advisor en cours d'exécution

L'`stop-suite-run`exemple suivant arrête une suite de tests Device Advisor actuellement en cours d'exécution avec l'ID de définition de suite et l'ID d'exécution de suite spécifiés.

```
aws iotdeviceadvisor stop-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-run-id nzlfyhaa18oa
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Arrêter l'exécution d'une suite de tests dans le Guide du développeur AWS IoT Core.

- Pour plus de détails sur l'API, reportez-vous StopSuiteRunà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter et modifier les balises existantes d'une ressource IoT Device Advisor

L'`tag-resource`exemple suivant ajoute et modifie les balises existantes d'une ressource Device Advisor avec l'ARN et les balises de ressource spécifiés. La ressource Device Advisor peut être un Suitedefinition-Arn ou un Suiterun-Arn.

```
aws iotdeviceadvisor tag-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/ba0uyjpg38ny \
    --tags '{"TagKey": "TagValue"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez TagResourcela référence des API AWS IoT et les types de ressources définis par AWS IoT Core Device Advisor dans la référence d'autorisation de service.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les balises existantes d'une ressource IoT Device Advisor

L'`untag-resource`exemple suivant supprime les balises existantes d'une ressource Device Advisor avec l'ARN de ressource et la clé de balise spécifiés. La ressource Device Advisor peut être un Suitedefinition-Arn ou un Suiterun-Arn.

```
aws iotdeviceadvisor untag-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny \
    --tag-keys "TagKey"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UntagResourcela référence des API AWS IoT et les types de ressources définis par AWS IoT Core Device Advisor dans la référence d'autorisation de service.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-suite-definition`

L'exemple de code suivant montre comment utiliser`update-suite-definition`.

AWS CLI

Exemple 1 : pour mettre à jour une suite de tests IoT Device Advisor

L'`update-suite-definition`exemple suivant met à jour une suite de tests Device Advisor dans l' AWS IoT avec l'ID de définition de suite et la configuration de définition de suite spécifiés.

```
aws iotdeviceadvisor update-suite-definition \
    --suite-definition-id 3hsn88h4p2g5 \
    --suite-definition-configuration '{ \
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": false, \
        "rootGroup": "{\"configuration\":{},\"tests\":[{\"name\":\"MQTT Connect\",
\"configuration\":{\"EXECUTION_TIMEOUT\":120},\"tests\":[{\"name\":\"MQTT_Connect\",
\"configuration\":{},\"test\":{\"id\":\"MQTT_Connect\",\"testCase\":null,\"version
\":\"0.0.0\"}}]}]}", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

Sortie :

```
{
    "suiteDefinitionId": "3hsn88h4p2g5",
    "suiteDefinitionName": "TestSuiteName",
```

```
    "suiteDefinitionVersion": "v3",
    "createdAt": "2022-11-17T14:15:56.830000-05:00",
    "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

Exemple 2 : pour mettre à jour une suite de tests de qualification IoT Device Advisor

L'`update-suite-definition`exemple suivant met à jour une suite de tests de qualification de Device Advisor dans l' AWS IoT avec l'ID de définition de suite et la configuration de définition de suite spécifiés.

```
aws iotdeviceadvisor update-suite-definition \
    --suite-definition-id txgsuolk2myj \
    --suite-definition-configuration '{
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": true, \
        "rootGroup": "", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

Sortie :

```
{
    "suiteDefinitionId": "txgsuolk2myj",
    "suiteDefinitionName": "TestSuiteName",
    "suiteDefinitionVersion": "v3",
    "createdAt": "2022-11-17T14:15:56.830000-05:00",
    "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

Pour plus d'informations, consultez UpdateSuiteDefinitionle manuel de référence des API AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateSuiteDefinitionà la section Référence des AWS CLI commandes.

AWS IoT data exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT data.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-thing-shadow`

L'exemple de code suivant montre comment utiliser`delete-thing-shadow`.

AWS CLI

Pour supprimer le document fantôme d'un appareil

L'`delete-thing-shadow`exemple suivant supprime l'intégralité du document fantôme pour le périphérique nomméMyRPi.

```
aws iot-data delete-thing-shadow \
    --thing-name MyRPi \
    "output.txt"
```

La commande ne produit aucune sortie à l'écran, mais `output.txt` contient des informations qui confirment la version et l'horodatage du document fantôme que vous avez supprimé.

```
{"version":2,"timestamp":1560270384}
```

Pour plus d'informations, consultez la section Utilisation des ombres dans le Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteThingShadowà la section Référence des AWS CLI commandes.

`get-thing-shadow`

L'exemple de code suivant montre comment utiliser`get-thing-shadow`.

AWS CLI

Pour obtenir un objet, un document fantôme

L'`get-thing-shadow`exemple suivant obtient le document fantôme correspondant à l'objet IoT spécifié.

```
aws iot-data get-thing-shadow \
    --thing-name MyRPi \
    output.txt
```

La commande ne produit aucune sortie sur l'écran, mais ce qui suit montre le contenu de `output.txt` :

```
{
  "state":{
    "reported":{
    "moisture":"low"
    }
  },
  "metadata":{
    "reported":{
      "moisture":{
        "timestamp":1560269319
      }
    }
  },
  "version":1,"timestamp":1560269405
}
```

Pour plus d'informations, consultez Device Shadow Service Data Flow dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetThingShadowà la section Référence des AWS CLI commandes.

`update-thing-shadow`

L'exemple de code suivant montre comment utiliser`update-thing-shadow`.

AWS CLI

Pour mettre à jour un objet, Shadow

L'`update-thing-shadow`exemple suivant modifie l'état actuel de l'ombre du périphérique pour l'objet spécifié et l'enregistre dans le fichier`output.txt`.

```
aws iot-data update-thing-shadow \
    --thing-name MyRPi \
    --payload "{"state":{"reported":{"moisture":"okay"}}}" \
    "output.txt"
```

La commande ne produit aucune sortie sur l'écran, mais ce qui suit montre le contenu de `output.txt` :

```
{
    "state": {
        "reported": {
            "moisture": "okay"
        }
    },
    "metadata": {
        "reported": {
            "moisture": {
                "timestamp": 1560270036
            }
        }
    },
    "version": 2,
    "timestamp": 1560270036
}
```

Pour plus d'informations, consultez Device Shadow Service Data Flow dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateThingShadowà la section Référence des AWS CLI commandes.

AWS IoT Events exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Events.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-put-message`

L'exemple de code suivant montre comment utiliser`batch-put-message`.

AWS CLI

Pour envoyer des messages (entrées) à AWS IoT Events

L'`batch-put-message`exemple suivant envoie un ensemble de messages au système AWS IoT Events. Chaque charge utile de message est transformée en entrée que vous spécifiez (`inputName`) et ingérée dans tous les détecteurs qui surveillent cette entrée. Si plusieurs messages sont envoyés, l'ordre dans lequel ils sont traités n'est pas garanti. Pour garantir la commande, vous devez envoyer les messages un par un et attendre une réponse positive.

```
aws iotevents-data batch-put-message \
    --cli-input-json file://highPressureMessage.json
```

Contenu de `highPressureMessage.json` :

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
```

```
}
```

Sortie :

```
{
    "BatchPutMessageErrorEntries": []
}
```

Pour plus d'informations, consultez BatchPutMessagela référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous BatchPutMessageà la section Référence des AWS CLI commandes.

`batch-update-detector`

L'exemple de code suivant montre comment utiliser`batch-update-detector`.

AWS CLI

Pour mettre à jour un détecteur (instance)

L'`batch-update-detector`exemple suivant met à jour l'état, les valeurs des variables et les paramètres de temporisation d'un ou de plusieurs détecteurs (instances) d'un modèle de détecteur spécifié.

```
aws iotevents-data batch-update-detector \
    --cli-input-json file://budFulton-A32.json
```

Contenu de budFulton-A32.json :

```
{
    "detectors": [
        {
            "messageId": "00001",
            "detectorModelName": "motorDetectorModel",
            "keyValue": "Fulton-A32",
            "state": {
                "stateName": "Normal",
                "variables": [
                    {
                        "name": "pressureThresholdBreached",
                        "value": "0"
```

```
                    }
                ],
                "timers": [
                ]
            }
        }
    ]
}
```

Sortie :

```
{
    "batchUpdateDetectorErrorEntries": []
}
```

Pour plus d'informations, consultez BatchUpdateDetectorla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous BatchUpdateDetectorà la section Référence des AWS CLI commandes.

`create-detector-model`

L'exemple de code suivant montre comment utiliser`create-detector-model`.

AWS CLI

Pour créer un modèle de détecteur

L'`create-detector-model`exemple suivant crée un modèle de détecteur dont la configuration est spécifiée par un fichier de paramètres.

```
aws iotevents create-detector-model  \
    --cli-input-json file://motorDetectorModel.json
```

Contenu de `motorDetectorModel.json` :

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
```

```
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
"$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached &gt;
1",
```

```
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "events": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "3"
                                    }
                                }
                            ]
                        },
                        {
                            "eventName": "Pressure Okay",
                            "condition": "$input.PressureInput.sensorData.pressure
 &lt;= 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached - 1"
                                    }
                                }
                            ]
                        }
                    ],
                    "transitionEvents": [
                        {
                            "eventName": "BackToNormal",
```

```
                            "condition": "$input.PressureInput.sensorData.pressure
 &lt;= 70 &amp;&amp; $variable.pressureThresholdBreached &lt;= 1",
                            "nextState": "Normal"
                        }
                    ]
                },
                "onExit": {
                    "events": [
                        {
                            "eventName": "Normal Pressure Restored",
                            "condition": "true",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
                            ]
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    },
    "key": "motorid",
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

Sortie :

```
{
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560796816.077,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560796816.077,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
```

```
    }
}
```

Pour plus d'informations, consultez CreateDetectorModella référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous CreateDetectorModelà la section Référence des AWS CLI commandes.

`create-input`

L'exemple de code suivant montre comment utiliser`create-input`.

AWS CLI

Pour créer une entrée

L'`create-input`exemple suivant crée une entrée.

```
aws iotevents create-input  \
    --cli-input-json file://pressureInput.json
```

Contenu de `pressureInput.json` :

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

Sortie :

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795312.542,
        "creationTime": 1560795312.542,
```

```
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

Pour plus d'informations, consultez CreateInputla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous CreateInputà la section Référence des AWS CLI commandes.

`delete-detector-model`

L'exemple de code suivant montre comment utiliser`delete-detector-model`.

AWS CLI

Pour supprimer un modèle de détecteur

L'`delete-detector-model`exemple suivant supprime le modèle de détecteur spécifié. Toutes les instances actives du modèle de détecteur sont également supprimées.

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteDetectorModella référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous DeleteDetectorModelà la section Référence des AWS CLI commandes.

`delete-input`

L'exemple de code suivant montre comment utiliser`delete-input`.

AWS CLI

Pour supprimer une entrée

L'`delete-input`exemple suivant supprime l'entrée spécifiée.

```
aws iotevents delete-input \
```

```
    --input-name PressureInput
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteInputla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous DeleteInputà la section Référence des AWS CLI commandes.

`describe-detector-model`

L'exemple de code suivant montre comment utiliser`describe-detector-model`.

AWS CLI

Pour obtenir des informations sur un modèle de détecteur

L'`describe-detector-model`exemple suivant affiche les détails du modèle de détecteur spécifié. Comme le `version` paramètre n'est pas spécifié, les informations relatives à la dernière version sont renvoyées.

```
aws iotevents describe-detector-model \
    --detector-model-name motorDetectorModel
```

Sortie :

```
{
    "detectorModel": {
        "detectorModelConfiguration": {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "key": "motorid",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        },
        "detectorModelDefinition": {
            "states": [
                {
                    "onInput": {
```

```
                        "transitionEvents": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value":
"$variable.pressureThresholdBreached + 3"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure > 70",
                                "nextState": "Dangerous"
                            }
                        ],
                        "events": []
                    },
                    "stateName": "Normal",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "init",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "0"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    },
                    "onExit": {
                        "events": []
                    }
                },
                {
                    "onInput": {
```

```
                        "transitionEvents": [
                            {
                                "eventName": "BackToNormal",
                                "actions": [],
                                "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
                                "nextState": "Normal"
                            }
                        ],
                        "events": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "3"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure > 70"
                            },
                            {
                                "eventName": "Pressure Okay",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value":
"$variable.pressureThresholdBreached - 1"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure <= 70"
                            }
                        ]
                    },
                    "stateName": "Dangerous",
                    "onEnter": {
```

```
                        "events": [
                            {
                                "eventName": "Pressure Threshold Breached",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                        }
                                    }
                                ],
                                "condition": "$variable.pressureThresholdBreached >
 1"
                            }
                        ]
                    },
                    "onExit": {
                        "events": [
                            {
                                "eventName": "Normal Pressure Restored",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    }
                }
            ],
            "initialStateName": "Normal"
        }
    }
}
```

Pour plus d'informations, consultez DescribeDetectorModella référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous DescribeDetectorModelà la section Référence des AWS CLI commandes.

`describe-detector`

L'exemple de code suivant montre comment utiliser`describe-detector`.

AWS CLI

Pour obtenir des informations sur un détecteur (instance).

L'`describe-detector`exemple suivant affiche les détails du détecteur spécifié (instance).

```
aws iotevents-data describe-detector \
    --detector-model-name motorDetectorModel \
    --key-value "Fulton-A32"
```

Sortie :

```
{
    "detector": {
        "lastUpdateTime": 1560797852.776,
        "creationTime": 1560797852.775,
        "state": {
            "variables": [
                {
                    "name": "pressureThresholdBreached",
                    "value": "3"
                }
            ],
            "stateName": "Dangerous",
            "timers": []
        },
        "keyValue": "Fulton-A32",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

Pour plus d'informations, consultez DescribeDetectorla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous DescribeDetectorà la section Référence des AWS CLI commandes.

`describe-input`

L'exemple de code suivant montre comment utiliser`describe-input`.

AWS CLI

Pour obtenir des informations sur une entrée

L'`describe-input`exemple suivant affiche les détails de l'entrée spécifiée.

```
aws iotevents describe-input \
    --input-name PressureInput
```

Sortie :

```
{
    "input": {
        "inputConfiguration": {
            "status": "ACTIVE",
            "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/
PressureInput",
            "lastUpdateTime": 1560795312.542,
            "creationTime": 1560795312.542,
            "inputName": "PressureInput",
            "inputDescription": "Pressure readings from a motor"
        },
        "inputDefinition": {
            "attributes": [
                {
                    "jsonPath": "sensorData.pressure"
                },
                {
                    "jsonPath": "motorid"
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez DescribeInputla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous DescribeInputà la section Référence des AWS CLI commandes.

`describe-logging-options`

L'exemple de code suivant montre comment utiliser`describe-logging-options`.

AWS CLI

Pour obtenir des informations sur les paramètres de journalisation

L'`describe-logging-options`exemple suivant récupère les paramètres actuels des options de journalisation des événements AWS IoT.

```
aws iotevents describe-logging-options
```

Sortie :

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

Pour plus d'informations, consultez DescribeLoggingOptionsla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous DescribeLoggingOptionsà la section Référence des AWS CLI commandes.

`list-detector-model-versions`

L'exemple de code suivant montre comment utiliser`list-detector-model-versions`.

AWS CLI

Pour obtenir des informations sur les versions d'un modèle de détecteur

L'`list-detector-model-versions`exemple suivant répertorie toutes les versions d'un modèle de détecteur. Seules les métadonnées associées à chaque version du modèle de détecteur sont renvoyées.

```
aws iotevents list-detector-model-versions \
```

```
    --detector-model-name motorDetectorModel
```

Sortie :

```
{
    "detectorModelVersionSummaries": [
        {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

Pour plus d'informations, consultez ListDetectorModelVersionsla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous ListDetectorModelVersionsà la section Référence des AWS CLI commandes.

`list-detector-models`

L'exemple de code suivant montre comment utiliser`list-detector-models`.

AWS CLI

Pour obtenir la liste de vos modèles de détecteurs

L'`list-detector-models`exemple suivant répertorie les modèles de détecteurs que vous avez créés. Seules les métadonnées associées à chaque modèle de détecteur sont renvoyées.

```
aws iotevents list-detector-models
```

Sortie :

```
{
```

```
    "detectorModelSummaries": [
        {
            "detectorModelName": "motorDetectorModel",
            "creationTime": 1552072424.212
            "detectorModelDescription": "Detect overpressure in a motor."
        }
    ]
}
```

Pour plus d'informations, consultez ListDetectorModelsla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous ListDetectorModelsà la section Référence des AWS CLI commandes.

`list-detectors`

L'exemple de code suivant montre comment utiliser`list-detectors`.

AWS CLI

Pour obtenir la liste des détecteurs pour un modèle de détecteur

L'`list-detectors`exemple suivant répertorie les détecteurs (les instances d'un modèle de détecteur) de votre compte.

```
aws iotevents-data list-detectors \
    --detector-model-name motorDetectorModel
```

Sortie :

```
{
    "detectorSummaries": [
        {
            "lastUpdateTime": 1558129925.2,
            "creationTime": 1552073155.527,
            "state": {
                "stateName": "Normal"
            },
            "keyValue": "Fulton-A32",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
```

```
    ]
}
```

Pour plus d'informations, consultez ListDetectorsla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous ListDetectorsà la section Référence des AWS CLI commandes.

`list-inputs`

L'exemple de code suivant montre comment utiliser`list-inputs`.

AWS CLI

Pour répertorier les entrées

L'`list-inputs`exemple suivant répertorie les entrées que vous avez créées dans votre compte.

```
aws iotevents list-inputs
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1551742986.768,
        "creationTime": 1551742986.768,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

Pour plus d'informations, consultez ListInputsla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous ListInputsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises attribuées à une ressource.

L'`list-tags-for-resource`exemple suivant répertorie les noms et valeurs des clés de balise que vous avez attribués à la ressource.

```
aws iotevents list-tags-for-resource \
    --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Sortie :

```
{
    "tags": [
        {
            "value": "motor",
            "key": "deviceType"
        }
    ]
}
```

Pour plus d'informations, consultez ListTagsForResourcela référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-logging-options`

L'exemple de code suivant montre comment utiliser`put-logging-options`.

AWS CLI

Pour définir les options de journalisation

L'`put-logging-options`exemple suivant définit ou met à jour les options de journalisation des AWS IoT Events. Si vous mettez à jour la valeur d'un `loggingOptions` field, it can take up to one minute for the change to take effect. Also, if you change the policy attached to the role you specified in the ``roleArn` champ (par exemple, pour corriger une politique non valide), la prise en compte de cette modification peut prendre jusqu'à cinq minutes.

```
aws iotevents put-logging-options \
    --cli-input-json file://logging-options.json
```

Contenu de `logging-options.json` :

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "level": "DEBUG",
        "enabled": true,
        "detectorDebugOptions": [
            {
                "detectorModelName": "motorDetectorModel",
                "keyValue": "Fulton-A32"
            }
        ]
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez PutLoggingOptionsla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous PutLoggingOptionsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une ressource

L'`tag-resource`exemple suivant ajoute ou modifie (si la clé existe `deviceType` déjà) la balise attachée à la ressource spécifiée.

```
aws iotevents tag-resource \
    --cli-input-json file://pressureInput.tag.json
```

Contenu de `pressureInput.tag.json` :

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tags": [
        {
            "key": "deviceType",
            "value": "motor"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez TagResourcela référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`untag-resource`exemple suivant supprime la balise portant le nom de clé spécifié de la ressource spécifiée.

```
aws iotevents untag-resource \
    --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \
    --tagkeys deviceType
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UntagResourcela référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-detector-model`

L'exemple de code suivant montre comment utiliser`update-detector-model`.

AWS CLI

Pour mettre à jour un modèle de détecteur

L'`update-detector-model`exemple suivant met à jour le modèle de détecteur spécifié. Les détecteurs (instances) générés par la version précédente sont supprimés puis recréés à mesure que de nouvelles entrées arrivent.

```
aws iotevents update-detector-model \
    --cli-input-json file://motorDetectorModel.update.json
```

Contenu de `motorDetectorModel.update.json` :

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure >
 70",
                            "actions": [
                                {
```

```
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached > 1",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "events": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure >
 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "3"
                                    }
                                }
                            ]
```

```
                        },
                        {
                            "eventName": "Pressure Okay",
                            "condition": "$input.PressureInput.sensorData.pressure
 <= 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached - 1"
                                    }
                                }
                            ]
                        }
                    ],
                    "transitionEvents": [
                        {
                            "eventName": "BackToNormal",
                            "condition": "$input.PressureInput.sensorData.pressure
 <= 70 && $variable.pressureThresholdBreached <= 1",
                            "nextState": "Normal"
                        }
                    ]
                },
                "onExit": {
                    "events": [
                        {
                            "eventName": "Normal Pressure Restored",
                            "condition": "true",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
                            ]
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
```

```
    },
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

Sortie :

```
{
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560799387.719,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560799387.719,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "2"
    }
}
```

Pour plus d'informations, consultez UpdateDetectorModella référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous UpdateDetectorModelà la section Référence des AWS CLI commandes.

`update-input`

L'exemple de code suivant montre comment utiliser`update-input`.

AWS CLI

Pour mettre à jour une entrée

L'`update-input`exemple suivant met à jour l'entrée spécifiée avec une nouvelle description et une nouvelle définition.

```
aws iotevents update-input \
    --cli-input-json file://pressureInput.json
```

Contenu de `pressureInput.json` :

```
{
```

```
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

Sortie :

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795976.458,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

Pour plus d'informations, consultez UpdateInputla référence de l'API AWS IoT Events.

- Pour plus de détails sur l'API, reportez-vous UpdateInputà la section Référence des AWS CLI commandes.

AWS IoT Events-Data exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Events-Data.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-put-message`

L'exemple de code suivant montre comment utiliser`batch-put-message`.

AWS CLI

Pour envoyer des messages (entrées) à AWS IoT Events

L'`batch-put-message`exemple suivant envoie un ensemble de messages au système AWS IoT Events. Chaque charge utile de message est transformée en entrée que vous spécifiez (`inputName`) et ingérée dans tous les détecteurs qui surveillent cette entrée. Si plusieurs messages sont envoyés, l'ordre dans lequel ils sont traités n'est pas garanti. Pour garantir la commande, vous devez envoyer les messages un par un et attendre une réponse positive.

```
aws iotevents-data batch-put-message \
    --cli-binary-format raw-in-base64-out \
    --cli-input-json file://highPressureMessage.json
```

Contenu de `highPressureMessage.json` :

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
}
```

Sortie :

```
{
    "BatchPutMessageErrorEntries": []
}
```

Pour plus d'informations, consultez BatchPutMessagele guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous BatchPutMessageà la section Référence des AWS CLI commandes.

`batch-update-detector`

L'exemple de code suivant montre comment utiliser`batch-update-detector`.

AWS CLI

Pour mettre à jour un détecteur (instance)

L'`batch-update-detector`exemple suivant met à jour l'état, les valeurs des variables et les paramètres de temporisation d'un ou de plusieurs détecteurs (instances) d'un modèle de détecteur spécifié.

```
aws iotevents-data batch-update-detector \
    --cli-input-json file://budFulton-A32.json
```

Contenu de budFulton-A32.json :

```
{
    "detectors": [
        {
            "messageId": "00001",
            "detectorModelName": "motorDetectorModel",
            "keyValue": "Fulton-A32",
            "state": {
                "stateName": "Normal",
                "variables": [
                    {
                        "name": "pressureThresholdBreached",
                        "value": "0"
                    }
                ],
                "timers": [
                ]
            }
        }
    ]
}
```

Sortie :

```
{
    "batchUpdateDetectorErrorEntries": []
}
```

Pour plus d'informations, consultez BatchUpdateDetectorle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous BatchUpdateDetectorà la section Référence des AWS CLI commandes.

`create-detector-model`

L'exemple de code suivant montre comment utiliser`create-detector-model`.

AWS CLI

Pour créer un modèle de détecteur

L'`create-detector-model`exemple suivant crée un modèle de détecteur.

```
aws iotevents create-detector-model \
    --cli-input-json file://motorDetectorModel.json
```

Contenu de `motorDetectorModel.json` :

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
```

```
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached &gt;
 1",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
```

```
                ]
            },
            "onInput": {
                "events": [
                    {
                        "eventName": "Overpressurized",
                        "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
                        "actions": [
                            {
                                "setVariable": {
                                    "variableName": "pressureThresholdBreached",
                                    "value": "3"
                                }
                            }
                        ]
                    },
                    {
                        "eventName": "Pressure Okay",
                        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70",
                        "actions": [
                            {
                                "setVariable": {
                                    "variableName": "pressureThresholdBreached",
                                    "value":
"$variable.pressureThresholdBreached - 1"
                                }
                            }
                        ]
                    }
                ],
                "transitionEvents": [
                    {
                        "eventName": "BackToNormal",
                        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 &amp;&amp; $variable.pressureThresholdBreached &lt;= 1",
                        "nextState": "Normal"
                    }
                ]
            },
            "onExit": {
                "events": [
                    {
```

```
                            "eventName": "Normal Pressure Restored",
                            "condition": "true",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
                            ]
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    },
    "key": "motorid",
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

Sortie :

```
{
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560796816.077,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560796816.077,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

Pour plus d'informations, consultez CreateDetectorModelle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous CreateDetectorModelà la section Référence des AWS CLI commandes.

`create-input`

L'exemple de code suivant montre comment utiliser`create-input`.

AWS CLI

Pour créer une entrée

L'`create-input`exemple suivant crée une entrée.

```
aws iotevents create-input \
    --cli-input-json file://pressureInput.json
```

Contenu de `pressureInput.json` :

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
                { "jsonPath": "sensorData.pressure" },
                { "jsonPath": "motorid" }
            ]
    }
}
```

Sortie :

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795312.542,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

Pour plus d'informations, consultez CreateInputle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous CreateInputà la section Référence des AWS CLI commandes.

`delete-detector-model`

L'exemple de code suivant montre comment utiliser`delete-detector-model`.

AWS CLI

Pour supprimer un modèle de détecteur

L'`delete-detector-model`exemple suivant supprime un modèle de détecteur. Toutes les instances actives du modèle de détecteur sont également supprimées.

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel*
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteDetectorModelle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous DeleteDetectorModelà la section Référence des AWS CLI commandes.

`delete-input`

L'exemple de code suivant montre comment utiliser`delete-input`.

AWS CLI

Pour supprimer une entrée

L'`delete-input`exemple suivant supprime une entrée.

```
aws iotevents delete-input \
    --input-name PressureInput
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteInputle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous DeleteInputà la section Référence des AWS CLI commandes.

`describe-detector-model`

L'exemple de code suivant montre comment utiliser`describe-detector-model`.

AWS CLI

Pour obtenir des informations sur un modèle de détecteur

L'`describe-detector-model`exemple suivant décrit un modèle de détecteur. Si le `version` paramètre n'est pas spécifié, la commande renvoie des informations sur la dernière version.

```
aws iotevents describe-detector-model \
    --detector-model-name motorDetectorModel
```

Sortie :

```
{
    "detectorModel": {
        "detectorModelConfiguration": {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "key": "motorid",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        },
        "detectorModelDefinition": {
            "states": [
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value":
 "$variable.pressureThresholdBreached + 3"
```

```
                                    }
                                }
                            ],
                            "condition":
"$input.PressureInput.sensorData.pressure > 70",
                            "nextState": "Dangerous"
                        }
                    ],
                    "events": []
                },
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName":
"pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ],
                            "condition": "true"
                        }
                    ]
                },
                "onExit": {
                    "events": []
                }
            },
            {
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "BackToNormal",
                            "actions": [],
                            "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
                            "nextState": "Normal"
                        }
                    ],
```

```
                        "events": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value": "3"
                                        }
                                    }
                                ],
                                "condition":
 "$input.PressureInput.sensorData.pressure > 70"
                            },
                            {
                                "eventName": "Pressure Okay",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value":
 "$variable.pressureThresholdBreached - 1"
                                        }
                                    }
                                ],
                                "condition":
 "$input.PressureInput.sensorData.pressure <= 70"
                            }
                        ]
                    },
                    "stateName": "Dangerous",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "Pressure Threshold Breached",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                        }
                                    }
```

```
                                ],
                                "condition": "$variable.pressureThresholdBreached >
 1"
                            }
                        ]
                    },
                    "onExit": {
                        "events": [
                            {
                                "eventName": "Normal Pressure Restored",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    }
                }
            ],
            "initialStateName": "Normal"
        }
    }
}
```

Pour plus d'informations, consultez DescribeDetectorModelle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous DescribeDetectorModelà la section Référence des AWS CLI commandes.

`describe-detector`

L'exemple de code suivant montre comment utiliser`describe-detector`.

AWS CLI

Pour obtenir des informations sur un détecteur (instance)

L'`describe-detector`exemple suivant renvoie des informations sur le détecteur spécifié (instance).

```
aws iotevents-data describe-detector \
    --detector-model-name motorDetectorModel \
    --key-value "Fulton-A32"
```

Sortie :

```
{
    "detector": {
        "lastUpdateTime": 1560797852.776,
        "creationTime": 1560797852.775,
        "state": {
            "variables": [
                {
                    "name": "pressureThresholdBreached",
                    "value": "3"
                }
            ],
            "stateName": "Dangerous",
            "timers": []
        },
        "keyValue": "Fulton-A32",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

Pour plus d'informations, consultez DescribeDetectorle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous DescribeDetectorà la section Référence des AWS CLI commandes.

`describe-input`

L'exemple de code suivant montre comment utiliser`describe-input`.

AWS CLI

Pour obtenir des informations sur une entrée

L'`describe-input`exemple suivant permet de récupérer les détails d'une entrée.

```
aws iotevents describe-input \
    --input-name PressureInput
```

Sortie :

```
{
    "input": {
        "inputConfiguration": {
            "status": "ACTIVE",
            "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/
PressureInput",
            "lastUpdateTime": 1560795312.542,
            "creationTime": 1560795312.542,
            "inputName": "PressureInput",
            "inputDescription": "Pressure readings from a motor"
        },
        "inputDefinition": {
            "attributes": [
                {
                    "jsonPath": "sensorData.pressure"
                },
                {
                    "jsonPath": "motorid"
                }
            ]
        }
    }
}
```

Pour plus d'informations, consultez DescribeInputle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous DescribeInputà la section Référence des AWS CLI commandes.

`describe-logging-options`

L'exemple de code suivant montre comment utiliser`describe-logging-options`.

AWS CLI

Pour obtenir des informations sur les paramètres de journalisation

L'`describe-logging-options`exemple suivant récupère les options actuelles de journalisation des AWS IoT Events.

```
aws iotevents describe-logging-options
```

Sortie :

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

Pour plus d'informations, consultez DescribeLoggingOptionsle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous DescribeLoggingOptionsà la section Référence des AWS CLI commandes.

`list-detector-model-versions`

L'exemple de code suivant montre comment utiliser`list-detector-model-versions`.

AWS CLI

Pour obtenir des informations sur les versions d'un modèle de détecteur

L'`list-detector-model-versions`exemple suivant répertorie toutes les versions d'un modèle de détecteur. Seules les métadonnées associées à chaque version du modèle de détecteur sont renvoyées.

```
aws iotevents list-detector-model-versions \
    --detector-model-name motorDetectorModel
```

Sortie :

```
{
    "detectorModelVersionSummaries": [
```

```
        {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

Pour plus d'informations, consultez ListDetectorModelVersionsle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous ListDetectorModelVersionsà la section Référence des AWS CLI commandes.

`list-detector-models`

L'exemple de code suivant montre comment utiliser`list-detector-models`.

AWS CLI

Pour obtenir la liste de vos modèles de détecteurs

L'`list-detector-models`exemple suivant répertorie les modèles de détecteurs que vous avez créés. Seules les métadonnées associées à chaque modèle de détecteur sont renvoyées.

```
aws iotevents list-detector-models
```

Sortie :

```
{
    "detectorModelSummaries": [
        {
            "detectorModelName": "motorDetectorModel",
            "creationTime": 1552072424.212
            "detectorModelDescription": "Detect overpressure in a motor."
        }
    ]
```

```
}
```

Pour plus d'informations, consultez ListDetectorModelsle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous ListDetectorModelsà la section Référence des AWS CLI commandes.

`list-detectors`

L'exemple de code suivant montre comment utiliser`list-detectors`.

AWS CLI

Pour obtenir la liste des détecteurs pour un modèle de détecteur

L'`list-detectors`exemple suivant répertorie les détecteurs (les instances d'un modèle de détecteur).

```
aws iotevents-data list-detectors \
    --detector-model-name motorDetectorModel
```

Sortie :

```
{
    "detectorSummaries": [
        {
            "lastUpdateTime": 1558129925.2,
            "creationTime": 1552073155.527,
            "state": {
                "stateName": "Normal"
            },
            "keyValue": "Fulton-A32",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

Pour plus d'informations, consultez ListDetectorsle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous ListDetectorsà la section Référence des AWS CLI commandes.

`list-inputs`

L'exemple de code suivant montre comment utiliser`list-inputs`.

AWS CLI

Pour répertorier les entrées

L'`list-inputs`exemple suivant répertorie les entrées que vous avez créées.

```
aws iotevents list-inputs
```

Sortie :

```
{
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1551742986.768,
    "creationTime": 1551742986.768,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
}
```

Pour plus d'informations, consultez ListInputsle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous ListInputsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises attribuées à une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises (métadonnées) que vous avez attribuées à la ressource.

```
aws iotevents list-tags-for-resource \
    --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Sortie :

```
{
    "tags": [
        {
            "value": "motor",
            "key": "deviceType"
        }
    ]
}
```

Pour plus d'informations, consultez ListTagsForResourcele guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-logging-options`

L'exemple de code suivant montre comment utiliser`put-logging-options`.

AWS CLI

Pour définir les options de journalisation

L'`list-tags-for-resource`exemple suivant définit ou met à jour les options de journalisation des AWS IoT Events. Si vous mettez à jour la valeur d'un `loggingOptions` champ, la modification prend jusqu'à une minute pour prendre effet. De même, si vous modifiez la politique associée au rôle que vous avez spécifié dans le `roleArn` champ (par exemple, pour corriger une politique non valide), il faut jusqu'à cinq minutes pour que cette modification prenne effet.

```
aws iotevents put-logging-options \
    --cli-input-json file://logging-options.json
```

Contenu de `logging-options.json` :

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "level": "DEBUG",
        "enabled": true,
```

```
        "detectorDebugOptions": [
            {
                "detectorModelName": "motorDetectorModel",
                "keyValue": "Fulton-A32"
            }
        ]
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez PutLoggingOptionsle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous PutLoggingOptionsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une ressource

L'`tag-resource`exemple suivant ajoute ou modifie les balises de la ressource donnée. Les balises sont des métadonnées qui peuvent être utilisées pour gérer une ressource.

```
aws iotevents tag-resource \
    --cli-input-json file://pressureInput.tag.json
```

Contenu de `pressureInput.tag.json` :

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tags": [
        {
            "key": "deviceType",
            "value": "motor"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez TagResourcele guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`untag-resource`exemple suivant supprime les balises spécifiées de la ressource.

```
aws iotevents untag-resource \
    --cli-input-json file://pressureInput.untag.json
```

Contenu de `pressureInput.untag.json` :

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tagKeys": [
            "deviceType"
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez UntagResourcele guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-detector-model`

L'exemple de code suivant montre comment utiliser`update-detector-model`.

AWS CLI

Pour mettre à jour un modèle de détecteur

L'`update-detector-model`exemple suivant met à jour un modèle de détecteur. Les détecteurs (instances) générés par la version précédente sont supprimés puis recréés à mesure que de nouvelles entrées arrivent.

```
aws iotevents update-detector-model \
    --cli-input-json file://motorDetectorModel.update.json
```

Contenu du motorDetectorModel fichier .update.json :

```
{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "0"
                  }
                }
              ]
            }
          ]
        },
        "onInput": {
          "transitionEvents": [
            {
              "eventName": "Overpressurized",
              "condition": "$input.PressureInput.sensorData.pressure > 70",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "$variable.pressureThresholdBreached + 3"
                  }
                }
```

```
              ],
              "nextState": "Dangerous"
            }
          ]
        }
      },
      {
        "stateName": "Dangerous",
        "onEnter": {
          "events": [
            {
              "eventName": "Pressure Threshold Breached",
              "condition": "$variable.pressureThresholdBreached > 1",
              "actions": [
                {
                  "sns": {
                    "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                  }
                }
              ]
            }
          ]
        },
        "onInput": {
          "events": [
            {
              "eventName": "Overpressurized",
              "condition": "$input.PressureInput.sensorData.pressure > 70",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "3"
                  }
                }
              ]
            },
            {
              "eventName": "Pressure Okay",
              "condition": "$input.PressureInput.sensorData.pressure <= 70",
              "actions": [
                {
                  "setVariable": {
```

```
                    "variableName": "pressureThresholdBreached",
                    "value": "$variable.pressureThresholdBreached - 1"
                  }
                }
              ]
            }
          ],
          "transitionEvents": [
            {
              "eventName": "BackToNormal",
              "condition": "$input.PressureInput.sensorData.pressure <= 70 &&
 $variable.pressureThresholdBreached <= 1",
              "nextState": "Normal"
            }
          ]
        },
        "onExit": {
          "events": [
            {
              "eventName": "Normal Pressure Restored",
              "condition": "true",
              "actions": [
                {
                  "sns": {
                    "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                  }
                }
              ]
            }
          ]
        }
      }
    ],
    "initialStateName": "Normal"
  },
  "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

Sortie :

```
{
    "detectorModelConfiguration": {
```

```
        "status": "ACTIVATING",
        "lastUpdateTime": 1560799387.719,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560799387.719,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "2"
    }
}
```

Pour plus d'informations, consultez UpdateDetectorModelle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous UpdateDetectorModelà la section Référence des AWS CLI commandes.

`update-input`

L'exemple de code suivant montre comment utiliser`update-input`.

AWS CLI

Pour mettre à jour une entrée

L'`update-input`exemple suivant met à jour une entrée.

```
aws iotevents update-input \
    --cli-input-json file://pressureInput.json
```

Contenu de `pressureInput.json` :

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
```

```
}
```

Sortie :

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795976.458,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

Pour plus d'informations, consultez UpdateInputle guide du développeur AWS IoT Events*.

- Pour plus de détails sur l'API, reportez-vous UpdateInputà la section Référence des AWS CLI commandes.

AWS IoT Greengrass exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Greengrass.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-role-to-group`

L'exemple de code suivant montre comment utiliser`associate-role-to-group`.

AWS CLI

Pour associer un rôle à un groupe Greengrass

L'`associate-role-to-group`exemple suivant associe le rôle IAM spécifié à un groupe Greengrass. Le rôle de groupe est utilisé par les fonctions et connecteurs Lambda locaux pour accéder AWS aux services. Par exemple, votre rôle de groupe peut accorder les autorisations requises pour l'intégration CloudWatch des journaux.

```
aws greengrass associate-role-to-group \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \
    --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

Sortie :

```
{
    "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

Pour plus d'informations, consultez la section Configurer le rôle de groupe dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous AssociateRoleToGroupà la section Référence des AWS CLI commandes.

`associate-service-role-to-account`

L'exemple de code suivant montre comment utiliser`associate-service-role-to-account`.

AWS CLI

Pour associer un rôle de service à votre AWS compte

L'`associate-service-role-to-account`exemple suivant associe un rôle de service IAM, spécifié par son ARN, à AWS IoT Greengrass dans AWS votre compte. Vous devez avoir préalablement créé le rôle de service dans IAM, et vous devez y associer un document de politique permettant à AWS IoT Greengrass d'assumer ce rôle.

```
aws greengrass associate-service-role-to-account \
    --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

Sortie :

```
{
    "AssociatedAt": "2019-06-25T18:12:45Z"
}
```

Pour plus d'informations, consultez Greengrass Service Role dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous AssociateServiceRoleToAccountà la section Référence des AWS CLI commandes.

`create-connector-definition-version`

L'exemple de code suivant montre comment utiliser`create-connector-definition-version`.

AWS CLI

Pour créer une version de définition de connecteur

L'`create-connector-definition-version`exemple suivant crée une version de définition de connecteur et l'associe à la définition de connecteur spécifiée. Tous les connecteurs d'une version définissent des valeurs pour leurs paramètres.

```
aws greengrass create-connector-definition-version \
    --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
    --connectors "[{\"Id\": \"MyTwilioNotificationsConnector\",
 \"ConnectorArn\": \"arn:aws:greengrass:us-west-2::/connectors/
TwilioNotifications/versions/2\", \"Parameters\": {\"TWILIO_ACCOUNT_SID
\": \"AC1a8d4204890840d7fc482aab38090d57\", \"TwilioAuthTokenSecretArn\":
 \"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-
ntSlp6\", \"TwilioAuthTokenSecretArn-ResourceId\": \"TwilioAuthToken\",
 \"DefaultFromPhoneNumber\": \"4254492999\"}}]"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/55d0052b-0d7d-44d6-b56f-21867215e118/versions/33f709a0-c825-49cb-9eea-
dc8964fbd635",
    "CreationTimestamp": "2019-06-24T20:46:30.134Z",
    "Id": "55d0052b-0d7d-44d6-b56f-21867215e118",
```

```
    "Version": "33f709a0-c825-49cb-9eea-dc8964fbd635"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateConnectorDefinitionVersionà la section Référence des AWS CLI commandes.

`create-connector-definition`

L'exemple de code suivant montre comment utiliser`create-connector-definition`.

AWS CLI

Pour créer une définition de connecteur

L'`create-connector-definition`exemple suivant crée une définition de connecteur et une version de définition de connecteur initiale. La version initiale contient un connecteur. Tous les connecteurs d'une version définissent des valeurs pour leurs paramètres.

```
aws greengrass create-connector-definition \
    --name MySNSConnector \
    --initial-version "{\"Connectors\": [{\"Id\":\"MySNSConnector\",\"ConnectorArn
\":\"arn:aws:greengrass:us-west-2::/connectors/SNS/versions/1\",\"Parameters\":
 {\"DefaultSNSArn\":\"arn:aws:sns:us-west-2:123456789012:GGConnectorTopic\"}}]}"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
    "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
    "Name": "MySNSConnector"
}
```

Pour plus d'informations, consultez Getting Started with Greengrass Connectors (CLI) dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateConnectorDefinitionà la section Référence des AWS CLI commandes.

`create-core-definition-version`

L'exemple de code suivant montre comment utiliser`create-core-definition-version`.

AWS CLI

Pour créer une version de définition de base

L'`create-core-definition-version`exemple suivant crée une version de définition de base et l'associe à la définition de base spécifiée. La version ne peut contenir qu'un seul noyau. Avant de créer un noyau, vous devez d'abord créer et approvisionner l'objet AWS IoT correspondant. Ce processus inclut les `iot` commandes suivantes, qui renvoient le `ThingArn` et `CertificateArn` requis pour la `create-core-definition-version` commande.

Créez l'objet AWS IoT correspondant à l'appareil principal :

```
aws iot create-thing \
    --thing-name "MyCoreDevice"
```

Sortie :

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
    "thingName": "MyCoreDevice",
    "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

Créez des clés publiques et privées ainsi que le certificat de périphérique principal pour l'objet. Cet exemple utilise la `create-keys-and-certificate` commande et nécessite des autorisations d'écriture sur le répertoire en cours. Vous pouvez également utiliser la `create-certificate-from-csr` commande.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myCore.cert.pem" \
    --public-key-outfile "myCore.public.key" \
```

```
    --private-key-outfile "myCore.private.key"
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

Créez une politique AWS IoT qui autorise `iot` et prend `greengrass` des mesures. Pour des raisons de simplicité, la politique suivante autorise les actions sur toutes les ressources, mais votre politique doit être plus restrictive.

```
aws iot create-policy \
    --policy-name "Core_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

Sortie :

```
{
    "policyName": "Core_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
```

```
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

Joignez la politique au certificat :

```
aws iot attach-policy \
    --policy-name "Core_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Cette commande ne produit aucun résultat.

Joignez l'objet au certificat :

```
aws iot attach-thing-principal \
    --thing-name "MyCoreDevice" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Cette commande ne produit aucun résultat.

Créez la version de définition de base :

```
aws greengrass create-core-definition-version \
    --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
    --cores "[{\"Id\":\"MyCoreDevice\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/MyCoreDevice\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\",\"SyncShadow\":true}]"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/3fdc1190-2ce5-44de-b98b-
eec8f9571014",
    "Version": "3fdc1190-2ce5-44de-b98b-eec8f9571014",
```

```
    "CreationTimestamp": "2019-09-18T00:15:09.838Z",
    "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

Pour plus d'informations, voir Configurer le cœur AWS IoT Greengrass dans le Guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateCoreDefinitionVersionà la section Référence des AWS CLI commandes.

`create-core-definition`

L'exemple de code suivant montre comment utiliser`create-core-definition`.

AWS CLI

Exemple 1 : pour créer une définition de base vide

L'`create-core-definition`exemple suivant crée une définition de base Greengrass vide (aucune version initiale). Avant que le noyau ne soit utilisable, vous devez utiliser la `create-core-definition-version` commande pour fournir les autres paramètres du noyau.

```
aws greengrass create-core-definition \
    --name cliGroup_Core
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
b5c08008-54cb-44bd-9eec-c121b04283b5",
    "CreationTimestamp": "2019-06-25T18:23:22.106Z",
    "Id": "b5c08008-54cb-44bd-9eec-c121b04283b5",
    "LastUpdatedTimestamp": "2019-06-25T18:23:22.106Z",
    "Name": "cliGroup_Core"
}
```

Exemple 2 : pour créer une définition de base avec une version initiale

L'`create-core-definition`exemple suivant crée une définition de base qui contient une version de définition de base initiale. La version ne peut contenir qu'un seul noyau. Avant de

créer un noyau, vous devez d'abord créer et approvisionner l'objet AWS IoT correspondant. Ce processus inclut les `iot` commandes suivantes, qui renvoient le `ThingArn` et `CertificateArn` requis pour la `create-core-definition` commande.

Créez l'objet AWS IoT correspondant à l'appareil principal :

```
aws iot create-thing \
    --thing-name "MyCoreDevice"
```

Sortie :

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
    "thingName": "MyCoreDevice",
    "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

Créez des clés publiques et privées ainsi que le certificat de périphérique principal pour l'objet. Cet exemple utilise la `create-keys-and-certificate` commande et nécessite des autorisations d'écriture sur le répertoire en cours. Vous pouvez également utiliser la `create-certificate-from-csr` commande.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myCore.cert.pem" \
    --public-key-outfile "myCore.public.key" \
    --private-key-outfile "myCore.private.key"
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
```

```
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

Créez une politique AWS IoT qui autorise `iot` et prend `greengrass` des mesures. Pour des raisons de simplicité, la politique suivante autorise les actions sur toutes les ressources, mais votre politique doit être plus restrictive.

```
aws iot create-policy \
    --policy-name "Core_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

Sortie :

```
{
    "policyName": "Core_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

Joignez la politique au certificat :

```
aws iot attach-policy \
    --policy-name "Core_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Cette commande ne produit aucun résultat.

Joignez l'objet au certificat :

```
aws iot attach-thing-principal \
    --thing-name "MyCoreDevice" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Cette commande ne produit aucun résultat.

Créez la définition de base :

```
aws greengrass create-core-definition \
    --name "MyCores" \
    --initial-version "{\"Cores\":[{\"Id\":\"MyCoreDevice\",\"ThingArn\":
\"arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice\",\"CertificateArn\":
\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\",\"SyncShadow\":true}]}"
```

Sortie :

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/
cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
    "Name": "MyCores",
    "LastUpdatedTimestamp": "2019-09-18T00:11:06.197Z",
    "LatestVersion": "cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
    "CreationTimestamp": "2019-09-18T00:11:06.197Z",
    "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

Pour plus d'informations, voir Configurer le cœur AWS IoT Greengrass dans le Guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateCoreDefinitionà la section Référence des AWS CLI commandes.

`create-deployment`

L'exemple de code suivant montre comment utiliser`create-deployment`.

AWS CLI

Pour créer un déploiement pour une version d'un groupe Greengrass

L'`create-deployment`exemple suivant déploie la version spécifiée d'un groupe Greengrass.

```
aws greengrass create-deployment \
    --deployment-type NewDeployment \
    --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
    --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

Sortie :

```
{
    "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/deployments/bfceb608-4e97-45bc-
af5c-460144270308",
    "DeploymentId": "bfceb608-4e97-45bc-af5c-460144270308"
}
```

Pour plus d'informations, consultez Getting Started with Connectors (CLI) dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentà la section Référence des AWS CLI commandes.

`create-device-definition-version`

L'exemple de code suivant montre comment utiliser`create-device-definition-version`.

AWS CLI

Pour créer une version de définition d'appareil

L'`create-device-definition-version`exemple suivant crée une version de définition de périphérique et l'associe à la définition de périphérique spécifiée. La version définit deux appareils. Avant de créer un appareil Greengrass, vous devez d'abord créer et approvisionner

l'objet AWS IoT correspondant. Ce processus inclut les `iot` commandes suivantes que vous devez exécuter pour obtenir les informations requises pour la commande Greengrass :

Créez l'objet AWS IoT correspondant à l'appareil :

```
aws iot create-thing \
    --thing-name "InteriorTherm"
```

Sortie :

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
    "thingName": "InteriorTherm",
    "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

Créez des clés publiques et privées ainsi que le certificat de périphérique correspondant à l'objet. Cet exemple utilise la `create-keys-and-certificate` commande et nécessite des autorisations d'écriture sur le répertoire en cours. Vous pouvez également utiliser la `create-certificate-from-csr` commande suivante :

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myDevice.cert.pem" \
    --public-key-outfile "myDevice.public.key" \
    --private-key-outfile "myDevice.private.key"
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
```

```
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

Créez une politique AWS IoT qui autorise `iot` et prend `greengrass` des mesures. Pour des raisons de simplicité, la politique suivante autorise des actions sur toutes les ressources, mais votre politique peut être plus restrictive :

```
aws iot create-policy \
    --policy-name "GG_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

Sortie :

```
{
    "policyName": "GG_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

Joignez la politique au certificat :

```
aws iot attach-policy \
    --policy-name "GG_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Attachez l'objet au certificat

```
aws iot attach-thing-principal \
    --thing-name "InteriorTherm" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Après avoir créé et configuré l'objet IoT comme indiqué ci-dessus, utilisez `ThingArn` les deux premières commandes `CertificateArn` de l'exemple suivant.

```
aws greengrass create-device-definition-version \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --devices "[{\"Id\":\"InteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/InteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\",
\"SyncShadow\":true},{\"Id\":\"ExteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\",
\"SyncShadow\":true}]"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "CreationTimestamp": "2019-09-11T00:15:09.838Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDeviceDefinitionVersionà la section Référence des AWS CLI commandes.

`create-device-definition`

L'exemple de code suivant montre comment utiliser`create-device-definition`.

AWS CLI

Pour créer une définition d'appareil

L'`create-device-definition`exemple suivant crée une définition de périphérique qui contient une version de définition de périphérique initiale. La version initiale définit deux appareils. Avant de créer un appareil Greengrass, vous devez d'abord créer et approvisionner l'objet AWS IoT correspondant. Ce processus inclut les `iot` commandes suivantes que vous devez exécuter pour obtenir les informations requises pour la commande Greengrass :

Créez l'objet AWS IoT correspondant à l'appareil :

```
aws iot create-thing \
    --thing-name "InteriorTherm"
```

Sortie :

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
    "thingName": "InteriorTherm",
    "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

Créez des clés publiques et privées ainsi que le certificat de périphérique correspondant à l'objet. Cet exemple utilise la `create-keys-and-certificate` commande et nécessite des autorisations d'écriture sur le répertoire en cours. Vous pouvez également utiliser la `create-certificate-from-csr` commande suivante :

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myDevice.cert.pem" \
    --public-key-outfile "myDevice.public.key" \
    --private-key-outfile "myDevice.private.key"
```

Sortie :

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
```

```
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

Créez une politique AWS IoT qui autorise `iot` et prend `greengrass` des mesures. Pour des raisons de simplicité, la politique suivante autorise des actions sur toutes les ressources, mais votre politique peut être plus restrictive :

```
aws iot create-policy \
    --policy-name "GG_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

Sortie :

```
{
    "policyName": "GG_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

Joignez la politique au certificat :

```
aws iot attach-policy \
    --policy-name "GG_Devices" \
```

```
    --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Attachez l'objet au certificat

```
aws iot attach-thing-principal \
    --thing-name "InteriorTherm" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Après avoir créé et configuré l'objet IoT comme indiqué ci-dessus, utilisez `ThingArn` les deux premières commandes `CertificateArn` de l'exemple suivant.

```
aws greengrass create-device-definition \
    --name "Sensors" \
    --initial-version "{\"Devices\":[{\"Id\":\"InteriorTherm
\",\"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/
InteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\",
\"SyncShadow\":true},{\"Id\":\"ExteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\",
\"SyncShadow\":true}]}"
```

Sortie :

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795",
    "Name": "Sensors",
    "LastUpdatedTimestamp": "2019-09-11T00:11:06.197Z",
    "LatestVersion": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
    "CreationTimestamp": "2019-09-11T00:11:06.197Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDeviceDefinitionà la section Référence des AWS CLI commandes.

`create-function-definition-version`

L'exemple de code suivant montre comment utiliser`create-function-definition-version`.

AWS CLI

Pour créer une version de la définition de fonction

L'`create-function-definition-version`exemple suivant crée une nouvelle version de la définition de fonction spécifiée. Cette version spécifie une fonction unique dont l'identifiant est`Hello-World-function`, autorise l'accès au système de fichiers et spécifie une taille de mémoire maximale et un délai d'expiration.

```
aws greengrass create-function-definition-version \
    --cli-input-json "{\"FunctionDefinitionId\": \"e626e8c9-3b8f-4bf3-9cdc-
d26ecdeb9fa3\",\"Functions\": [{\"Id\": \"Hello-World-function\", \"FunctionArn\":
 \""arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld_Counter:gghw-alias\",
\"FunctionConfiguration\": {\"Environment\": {\"AccessSysfs\": true},\"Executable\":
 \"greengrassHelloWorldCounter.function_handler\",\"MemorySize\": 16000,\"Pinned\":
 false,\"Timeout\": 25}}]}"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3/
versions/74abd1cc-637e-4abe-8684-9a67890f4043",
    "CreationTimestamp": "2019-06-25T22:03:43.376Z",
    "Id": "e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3",
    "Version": "74abd1cc-637e-4abe-8684-9a67890f4043"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFunctionDefinitionVersionà la section Référence des AWS CLI commandes.

`create-function-definition`

L'exemple de code suivant montre comment utiliser`create-function-definition`.

AWS CLI

Pour créer une définition de fonction Lambda

L'`create-function-definition`exemple suivant crée une définition de fonction Lambda et une version initiale en fournissant une liste de fonctions Lambda (dans ce cas, une liste d'une seule fonction nomméeTempMonitorFunction) et leurs configurations. Avant de créer la définition de fonction, vous avez besoin de l'ARN de la fonction Lambda. Pour créer la fonction et son alias, utilisez les `publish-version` commandes Lambda `create-function` et. La `create-function` commande Lambda nécessite l'ARN du rôle d'exécution, même si AWS IoT Greengrass n'utilise pas ce rôle car les autorisations sont spécifiées dans le rôle de groupe Greengrass. Vous pouvez utiliser la `create-role` commande IAM pour créer un rôle vide afin d'obtenir un ARN à utiliser avec celui de Lambda `create-function` ou vous pouvez utiliser un rôle d'exécution existant.

```
aws greengrass create-function-definition \
    --name MyGreengrassFunctions \
    --initial-version "{\"Functions\": [{\"Id\": \"TempMonitorFunction\",
 \"FunctionArn\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"FunctionConfiguration
\": {\"Executable\": \"temp_monitor.function_handler\", \"MemorySize\": 16000,
\"Timeout\": 5}}]}"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/3b0d0080-87e7-48c6-b182-503ec743a08b",
    "CreationTimestamp": "2019-06-19T22:24:44.585Z",
    "Id": "3b0d0080-87e7-48c6-b182-503ec743a08b",
    "LastUpdatedTimestamp": "2019-06-19T22:24:44.585Z",
    "LatestVersion": "67f918b9-efb4-40b0-b87c-de8c9faf085b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/versions/67f918b9-
efb4-40b0-b87c-de8c9faf085b",
    "Name": "MyGreengrassFunctions"
}
```

Pour plus d'informations, consultez Comment configurer l'accès aux ressources locales à l'aide de l'interface de ligne de AWS commande dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateFunctionDefinitionà la section Référence des AWS CLI commandes.

`create-group-certificate-authority`

L'exemple de code suivant montre comment utiliser`create-group-certificate-authority`.

AWS CLI

Pour créer une autorité de certification (CA) pour un groupe

L'`create-group-certificate-authority`exemple suivant crée ou fait pivoter une autorité de certification pour le groupe spécifié.

```
aws greengrass create-group-certificate-authority \
    --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
```

Sortie :

```
{
    "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEcc5"
}
```

Pour plus d'informations, consultez AWS IoT Greengrass Security dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateGroupCertificateAuthorityà la section Référence des AWS CLI commandes.

`create-group-version`

L'exemple de code suivant montre comment utiliser`create-group-version`.

AWS CLI

Pour créer une version d'un groupe Greengrass

L'`create-group-version`exemple suivant crée une version de groupe et l'associe au groupe spécifié. La version fait référence aux versions de base, de ressource, de connecteur, de fonction

et d'abonnement qui contiennent les entités à inclure dans cette version de groupe. Vous devez créer ces entités avant de pouvoir créer la version du groupe.

Pour créer une définition de ressource avec une version initiale, utilisez la `create-resource-definition` commande.Pour créer une définition de connecteur avec une version initiale, utilisez la `create-connector-definition` commande.Pour créer une définition de fonction avec une version initiale, utilisez la `create-function-definition` commande.Pour créer une définition d'abonnement avec une version initiale, utilisez la commande.Pour `create-subscription-definition` récupérer l'ARN de la dernière version de définition de base, utilisez la `get-group-version` commande et spécifiez l'ID de la dernière version du groupe.

```
aws greengrass create-group-version \
    --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
    --core-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/6a630442-8708-4838-ad36-eb98849d975e/
versions/6c87151b-1fb4-4cb2-8b31-6ee715d8f8ba" \
    --resource-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1" \
    --connector-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/55d0052b-0d7d-44d6-b56f-21867215e118/
versions/78a3331b-895d-489b-8823-17b4f9f418a0" \
    --function-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/
versions/67f918b9-efb4-40b0-b87c-de8c9faf085b" \
    --subscription-definition-version-arn "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/definition/subscriptions/9d611d57-5d5d-44bd-a3b4-
feccbdd69112/versions/aa645c47-ac90-420d-9091-8c7ffa4f103f"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/
ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/versions/e10b0459-4345-4a09-88a4-1af1f5d34638",
    "CreationTimestamp": "2019-06-20T18:42:47.020Z",
    "Id": "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca",
    "Version": "e10b0459-4345-4a09-88a4-1af1f5d34638"
}
```

Pour plus d'informations, consultez la section Présentation du modèle d'objet du groupe AWS IoT Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateGroupVersionà la section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour créer un groupe Greeengrass

L'`create-group`exemple suivant crée un groupe nommé`cli-created-group`.

```
aws greengrass create-group \
    --name cli-created-group
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/4e22bd92-898c-436b-ade5-434d883ff749",
    "CreationTimestamp": "2019-06-25T18:07:17.688Z",
    "Id": "4e22bd92-898c-436b-ade5-434d883ff749",
    "LastUpdatedTimestamp": "2019-06-25T18:07:17.688Z",
    "Name": "cli-created-group"
}
```

Pour plus d'informations, consultez la section Présentation du modèle d'objet du groupe AWS IoT Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`create-logger-definition-version`

L'exemple de code suivant montre comment utiliser`create-logger-definition-version`.

AWS CLI

Pour créer une version de définition de l'enregistreur

L'`create-logger-definition-version`exemple suivant crée une version de définition d'enregistreur et l'associe à une définition d'enregistreur. La version définit quatre configurations de journalisation : 1) les journaux des composants système sur le système de fichiers du périphérique principal, 2) les journaux des fonctions Lambda définis par l'utilisateur sur le système de fichiers du périphérique principal, 3) les journaux des composants du système dans CloudWatch Amazon Logs et 4) les journaux des fonctions Lambda définis par l'utilisateur dans Amazon Logs. CloudWatch Remarque : pour l'intégration CloudWatch des journaux, votre rôle de groupe doit accorder les autorisations appropriées.

```
aws greengrass create-logger-definition-version \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
    --loggers "[{\"Id\":\"1\",\"Component\":\"GreengrassSystem\",\"Level\":\"ERROR
\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":\"2\",\"Component\":\"Lambda
\",\"Level\":\"INFO\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":\"3\",
\"Component\":\"GreengrassSystem\",\"Level\":\"WARN\",\"Type\":\"AWSCloudWatch\"},
{\"Id\":\"4\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":\"AWSCloudWatch
\"}]"
```

Sortie :

```
{
 "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/loggers/
a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/49aedb1e-01a3-4d39-9871-3a052573f1ea",
 "Version": "49aedb1e-01a3-4d39-9871-3a052573f1ea",
 "CreationTimestamp": "2019-07-24T00:04:48.523Z",
 "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
}
```

Pour plus d'informations, consultez la section Surveillance à AWS l'aide des journaux IoT Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateLoggerDefinitionVersionà la section Référence des AWS CLI commandes.

`create-logger-definition`

L'exemple de code suivant montre comment utiliser`create-logger-definition`.

AWS CLI

Pour créer une définition d'enregistreur

L'`create-logger-definition`exemple suivant crée une définition d'enregistreur qui contient une version de définition d'enregistreur initiale. La version initiale définit trois configurations de journalisation : 1) les journaux des composants du système sur le système de fichiers du périphérique principal, 2) les journaux des fonctions Lambda définis par l'utilisateur sur le système de fichiers du périphérique principal et 3) les journaux des fonctions Lambda définis par l'utilisateur dans Amazon Logs. CloudWatch Remarque : pour l'intégration CloudWatch des journaux, votre rôle de groupe doit accorder les autorisations appropriées.

```
aws greengrass create-logger-definition \
    --name "LoggingConfigs" \
    --initial-version "{\"Loggers\":[{\"Id\":\"1\",\"Component\":\"GreengrassSystem
\",\"Level\":\"ERROR\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":
\"2\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Space\":10240,\"Type\":
\"FileSystem\"},{\"Id\":\"3\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":
\"AWSCloudWatch\"}]}"
```

Sortie :

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/de1d9854-1588-4525-
b25e-b378f60f2322",
    "Name": "LoggingConfigs",
    "LastUpdatedTimestamp": "2019-07-23T23:52:17.165Z",
    "LatestVersion": "de1d9854-1588-4525-b25e-b378f60f2322",
    "CreationTimestamp": "2019-07-23T23:52:17.165Z",
    "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
}
```

Pour plus d'informations, consultez la section Surveillance à AWS l'aide des journaux IoT Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateLoggerDefinitionà la section Référence des AWS CLI commandes.

`create-resource-definition-version`

L'exemple de code suivant montre comment utiliser`create-resource-definition-version`.

AWS CLI

Pour créer une version d'une définition de ressource

L'`create-resource-definition-version`exemple suivant crée une nouvelle version d'un TwilioAuthToken.

```
aws greengrass create-resource-definition-version \
    --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \
    --resources "[{\"Id\": \"TwilioAuthToken\",\"Name\": \"MyTwilioAuthToken
\",\"ResourceDataContainer\": {\"SecretsManagerSecretResourceData\": {\"ARN\":
 \"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-
ntSlp6\"}}}]"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/b3bcada0-5fb6-42df-
bf0b-1ee4f15e769e",
    "CreationTimestamp": "2019-06-24T21:17:25.623Z",
    "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
    "Version": "b3bcada0-5fb6-42df-bf0b-1ee4f15e769e"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateResourceDefinitionVersionà la section Référence des AWS CLI commandes.

`create-resource-definition`

L'exemple de code suivant montre comment utiliser`create-resource-definition`.

AWS CLI

Pour créer une définition de ressource

L'`create-resource-definition`exemple suivant crée une définition de ressource qui contient une liste de ressources à utiliser dans un groupe Greengrass. Dans cet exemple, une version initiale de la définition de ressource est incluse en fournissant une liste de ressources. La liste inclut une ressource pour un jeton d'autorisation Twilio et l'ARN pour un secret stocké dans AWS Secrets Manager. Vous devez créer le secret avant de pouvoir créer la définition de ressource.

```
aws greengrass create-resource-definition \
    --name MyGreengrassResources \
    --initial-version "{\"Resources\": [{\"Id\": \"TwilioAuthToken
\",\"Name\": \"MyTwilioAuthToken\",\"ResourceDataContainer\":
 {\"SecretsManagerSecretResourceData\": {\"ARN\": \"arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-TwilioAuthToken-ntSlp6\"}}}]}"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
    "CreationTimestamp": "2019-06-19T21:51:28.212Z",
    "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
    "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
    "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/a5f94d0b-
f6bc-40f4-bb78-7a1c5fe13ba1",
    "Name": "MyGreengrassResources"
}
```

Pour plus d'informations, consultez Comment configurer l'accès aux ressources locales à l'aide de l'interface de ligne de AWS commande dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateResourceDefinitionà la section Référence des AWS CLI commandes.

`create-software-update-job`

L'exemple de code suivant montre comment utiliser`create-software-update-job`.

AWS CLI

Pour créer une tâche de mise à jour logicielle pour un noyau

L'`create-software-update-job`exemple suivant crée une tâche de mise à jour over-the-air (OTA) pour mettre à jour le logiciel AWS IoT Greengrass Core sur le noyau dont le nom est. `MyFirstGroup_Core` Cette commande nécessite un rôle IAM qui autorise l'accès aux packages de mises à jour logicielles dans Amazon S3 et qui est inclus en `iot.amazonaws.com` tant qu'entité de confiance.

```
aws greengrass create-software-update-job \
    --update-targets-architecture armv7l \
    --update-targets [\"arn:aws:iot:us-west-2:123456789012:thing/MyFirstGroup_Core
\"] \
    --update-targets-operating-system raspbian \
    --software-to-update core \
    --s3-url-signer-role arn:aws:iam::123456789012:role/OTA_signer_role \
    --update-agent-log-level WARN
```

Sortie :

```
{
    "IotJobId": "GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",
    "IotJobArn": "arn:aws:iot:us-west-2:123456789012:job/
GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",
    "PlatformSoftwareVersion": "1.9.3"
}
```

Pour plus d'informations, consultez les mises à jour OTA du logiciel AWS IoT Greengrass Core dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateSoftwareUpdateJobà la section Référence des AWS CLI commandes.

`create-subscription-definition-version`

L'exemple de code suivant montre comment utiliser`create-subscription-definition-version`.

AWS CLI

Pour créer une nouvelle version d'une définition d'abonnement

L'`create-subscription-definition-version`exemple suivant crée une nouvelle version d'une définition d'abonnement contenant trois abonnements : une notification de déclenchement, une entrée de température et un état de sortie.

```
aws greengrass create-subscription-definition-version \
    --subscription-definition-id "9d611d57-5d5d-44bd-a3b4-feccbdd69112" \
    --subscriptions "[{\"Id\": \"TriggerNotification\", \"Source\":
 \"arn:aws:lambda:us-west-2:123456789012:function:TempMonitor:GG_TempMonitor
```

```
\", \"Subject\": \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2::/
connectors/TwilioNotifications/versions/1\"},{\"Id\": \"TemperatureInput\", \"Source
\": \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\"},{\"Id\": \"OutputStatus
\", \"Source\": \"arn:aws:greengrass:us-west-2::/connectors/TwilioNotifications/
versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/7b65dfae-50b6-4d0f-
b3e0-27728bfb0620",
    "CreationTimestamp": "2019-06-24T21:21:33.837Z",
    "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
    "Version": "7b65dfae-50b6-4d0f-b3e0-27728bfb0620"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSubscriptionDefinitionVersionà la section Référence des AWS CLI commandes.

`create-subscription-definition`

L'exemple de code suivant montre comment utiliser`create-subscription-definition`.

AWS CLI

Pour créer une définition d'abonnement

L'`create-subscription-definition`exemple suivant crée une définition d'abonnement et spécifie sa version initiale. La version initiale contient trois abonnements : un pour le thème MQTT auquel le connecteur est abonné, un pour autoriser une fonction à recevoir des relevés de température provenant de l' AWS IoT et un pour permettre à l' AWS IoT de recevoir des informations d'état du connecteur. L'exemple fournit l'ARN de l'alias de fonction Lambda créé précédemment à l'aide de la commande Lambda. `create-alias`

```
aws greengrass create-subscription-definition \
    --initial-version "{\"Subscriptions\": [{\"Id\":
 \"TriggerNotification\", \"Source\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"Subject\":
 \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2::/connectors/
```

```
TwilioNotifications/versions/1\"},{\"Id\": \"TemperatureInput\", \"Source\":
 \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\"},{\"Id\": \"OutputStatus
\", \"Source\": \"arn:aws:greengrass:us-west-2::/connectors/TwilioNotifications/
versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]}"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112",
    "CreationTimestamp": "2019-06-19T22:34:26.677Z",
    "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
    "LastUpdatedTimestamp": "2019-06-19T22:34:26.677Z",
    "LatestVersion": "aa645c47-ac90-420d-9091-8c7ffa4f103f",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/aa645c47-
ac90-420d-9091-8c7ffa4f103f"
}
```

Pour plus d'informations, consultez Getting Started with Connectors (CLI) dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous CreateSubscriptionDefinitionà la section Référence des AWS CLI commandes.

`delete-connector-definition`

L'exemple de code suivant montre comment utiliser`delete-connector-definition`.

AWS CLI

Pour supprimer une définition de connecteur

L'`delete-connector-definition`exemple suivant supprime la définition du connecteur Greengrass spécifiée. Si vous supprimez une définition de connecteur utilisée par un groupe, ce groupe ne peut pas être déployé correctement.

```
aws greengrass delete-connector-definition \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteConnectorDefinitionà la section Référence des AWS CLI commandes.

`delete-core-definition`

L'exemple de code suivant montre comment utiliser`delete-core-definition`.

AWS CLI

Pour supprimer une définition de base

L'`delete-core-definition`exemple suivant supprime la définition de base de Greengrass spécifiée, y compris toutes les versions. Si vous supprimez un noyau associé à un groupe Greengrass, ce groupe ne pourra pas être déployé correctement.

```
aws greengrass delete-core-definition \
    --core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteCoreDefinitionà la section Référence des AWS CLI commandes.

`delete-device-definition`

L'exemple de code suivant montre comment utiliser`delete-device-definition`.

AWS CLI

Pour supprimer une définition d'appareil

L'`delete-device-definition`exemple suivant supprime la définition de périphérique spécifiée, y compris toutes ses versions. Si vous supprimez une version de définition d'appareil utilisée par une version de groupe, la version de groupe ne peut pas être déployée correctement.

```
aws greengrass delete-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteDeviceDefinitionà la section Référence des AWS CLI commandes.

`delete-function-definition`

L'exemple de code suivant montre comment utiliser`delete-function-definition`.

AWS CLI

Pour supprimer une définition de fonction

L'`delete-function-definition`exemple suivant supprime la définition de fonction Greengrass spécifiée. Si vous supprimez une définition de fonction utilisée par un groupe, ce groupe ne peut pas être déployé correctement.

```
aws greengrass delete-function-definition \
    --function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteFunctionDefinitionà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe

L'`delete-group`exemple suivant supprime le groupe Greengrass spécifié.

```
aws greengrass delete-group \
    --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`delete-logger-definition`

L'exemple de code suivant montre comment utiliser`delete-logger-definition`.

AWS CLI

Pour supprimer une définition d'enregistreur

L'`delete-logger-definition`exemple suivant supprime la définition d'enregistreur spécifiée, y compris toutes les versions de définition d'enregistreur. Si vous supprimez une version de définition d'enregistreur utilisée par une version de groupe, la version de groupe ne peut pas être déployée correctement.

```
aws greengrass delete-logger-definition \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance à AWS l'aide des journaux IoT Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous DeleteLoggerDefinitionà la section Référence des AWS CLI commandes.

`delete-resource-definition`

L'exemple de code suivant montre comment utiliser`delete-resource-definition`.

AWS CLI

Pour supprimer une définition de ressource

L'`delete-resource-definition`exemple suivant supprime la définition de ressource spécifiée, y compris toutes les versions de ressources. Si vous supprimez une définition de ressource utilisée par un groupe, celui-ci ne peut pas être déployé correctement.

```
aws greengrass delete-resource-definition \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteResourceDefinitionà la section Référence des AWS CLI commandes.

`delete-subscription-definition`

L'exemple de code suivant montre comment utiliser`delete-subscription-definition`.

AWS CLI

Pour supprimer une définition d'abonnement

L'`delete-subscription-definition`exemple suivant supprime la définition d'abonnement Greengrass spécifiée. Si vous supprimez un abonnement utilisé par un groupe, celui-ci ne peut pas être déployé correctement.

```
aws greengrass delete-subscription-definition \
    --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriptionDefinitionà la section Référence des AWS CLI commandes.

`disassociate-role-from-group`

L'exemple de code suivant montre comment utiliser`disassociate-role-from-group`.

AWS CLI

Pour dissocier le rôle d'un groupe Greengrass

L'`disassociate-role-from-group`exemple suivant dissocie le rôle IAM du groupe Greengrass spécifié.

```
aws greengrass disassociate-role-from-group \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Sortie :

```
{
    "DisassociatedAt": "2019-09-10T20:05:49Z"
```

```
}
```

Pour plus d'informations, consultez la section Configurer le rôle de groupe dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous DisassociateRoleFromGroupà la section Référence des AWS CLI commandes.

`disassociate-service-role-from-account`

L'exemple de code suivant montre comment utiliser`disassociate-service-role-from-account`.

AWS CLI

Pour dissocier un rôle de service de votre compte AWS

L'`disassociate-service-role-from-account`exemple suivant supprime le rôle de service associé à votre AWS compte. Si vous n'utilisez le rôle de service dans aucune AWS région, utilisez la `delete-role-policy` commande pour détacher la politique `AWSGreengrassResourceAccessRolePolicy` gérée du rôle, puis utilisez la `delete-role` commande pour supprimer le rôle.

```
aws greengrass disassociate-service-role-from-account
```

Sortie :

```
{
    "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

Pour plus d'informations, consultez Greengrass Service Role dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous DisassociateServiceRoleFromAccountà la section Référence des AWS CLI commandes.

`get-associated-role`

L'exemple de code suivant montre comment utiliser`get-associated-role`.

AWS CLI

Pour obtenir le rôle associé à un groupe Greengrass

L'`get-associated-role`exemple suivant obtient le rôle IAM associé au groupe Greengrass spécifié. Le rôle de groupe est utilisé par les fonctions et connecteurs Lambda locaux pour accéder AWS aux services.

```
aws greengrass get-associated-role \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Sortie :

```
{
    "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",
    "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

Pour plus d'informations, consultez la section Configurer le rôle de groupe dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous GetAssociatedRoleà la section Référence des AWS CLI commandes.

`get-bulk-deployment-status`

L'exemple de code suivant montre comment utiliser`get-bulk-deployment-status`.

AWS CLI

Pour vérifier l'état de votre déploiement en masse

L'`get-bulk-deployment-status`exemple suivant récupère les informations d'état pour l'opération de déploiement en bloc spécifiée. Dans cet exemple, le fichier qui a spécifié les groupes à déployer possède un enregistrement d'entrée non valide.

```
aws greengrass get-bulk-deployment-status \
    --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Sortie :

```
{
    "BulkDeploymentMetrics": {
        "InvalidInputRecords": 1,
        "RecordsProcessed": 1,
        "RetryAttempts": 0
    },
    "BulkDeploymentStatus": "Completed",
    "CreatedAt": "2019-06-25T16:11:33.265Z",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Créer des déploiements groupés pour les groupes dans le Guide du AWS développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous GetBulkDeploymentStatusà la section Référence des AWS CLI commandes.

`get-connectivity-info`

L'exemple de code suivant montre comment utiliser`get-connectivity-info`.

AWS CLI

Pour obtenir les informations de connectivité d'un noyau Greengrass

L'`get-connectivity-info`exemple suivant montre les points de terminaison que les appareils peuvent utiliser pour se connecter au cœur Greengrass spécifié. Les informations de connectivité sont une liste d'adresses IP ou de noms de domaine, avec les numéros de port correspondants et les métadonnées facultatives définies par le client.

```
aws greengrass get-connectivity-info \
    --thing-name "MyGroup_Core"
```

Sortie :

```
{
    "ConnectivityInfo": [
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "127.0.0.1",
            "Id": "AUTOIP_127.0.0.1_0"
```

```
        },
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "192.168.1.3",
            "Id": "AUTOIP_192.168.1.3_1"
        },
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "::1",
            "Id": "AUTOIP_::1_2"
        },
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "fe80::1e69:ed93:f5b:f6d",
            "Id": "AUTOIP_fe80::1e69:ed93:f5b:f6d_3"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetConnectivityInfoà la section Référence des AWS CLI commandes.

`get-connector-definition-version`

L'exemple de code suivant montre comment utiliser`get-connector-definition-version`.

AWS CLI

Pour récupérer des informations sur une version spécifique d'une définition de connecteur

L'`get-connector-definition-version`exemple suivant extrait des informations sur la version spécifiée de la définition de connecteur spécifiée. Pour récupérer toutes les versions de la définition du connecteur, utilisez la `list-connector-definition-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée à la définition du connecteur, utilisez la `get-connector-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-connector-definition-version \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \
    --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-c7c2-4a26-
a7e2-7bf478ea2623",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Definition": {
        "Connectors": [
            {
                "ConnectorArn": "arn:aws:greengrass:us-west-2::/connectors/SNS/
versions/1",
                "Id": "MySNSConnector",
                "Parameters": {
                    "DefaultSNSArn": "arn:aws:sns:us-
west-2:123456789012:GGConnectorTopic"
                }
            }
        ]
    },
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
}
```

Pour plus d'informations, consultez la section Intégrer aux services et aux protocoles à l'aide des connecteurs Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous GetConnectorDefinitionVersionà la section Référence des AWS CLI commandes.

`get-connector-definition`

L'exemple de code suivant montre comment utiliser`get-connector-definition`.

AWS CLI

Pour récupérer des informations sur la définition d'un connecteur

L'`get-connector-definition`exemple suivant extrait des informations sur la définition de connecteur spécifiée. Pour récupérer les définitions IDs de vos connecteurs, utilisez la `list-connector-definitions` commande.

```
aws greengrass get-connector-definition \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
    "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
    "Name": "MySNSConnector",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Intégrer aux services et aux protocoles à l'aide des connecteurs Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous GetConnectorDefinitionà la section Référence des AWS CLI commandes.

`get-core-definition-version`

L'exemple de code suivant montre comment utiliser`get-core-definition-version`.

AWS CLI

Pour récupérer des informations sur une version spécifique de la définition de base de Greengrass

L'`get-core-definition-version`exemple suivant extrait des informations sur la version spécifiée de la définition de base spécifiée. Pour récupérer toutes les versions de la définition de base, utilisez la `list-core-definition-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée à la définition de base, utilisez la `get-core-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-core-definition-version \
```

```
    --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"  \
    --core-definition-version-id "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-fd9d-4312-a8fd-ffa9404a20e0",
    "CreationTimestamp": "2019-06-18T16:21:21.351Z",
    "Definition": {
        "Cores": [
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/928dea7b82331b47c3ff77b0e763fc5e64e2f7c884e6ef391baed9b6b8e21b45",
                "Id": "1a39aac7-0885-4417-91f6-23e4cea6c511",
                "SyncShadow": false,
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
GGGroup4Pi3_Core"
            }
        ]
    },
    "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
    "Version": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
}
```

- Pour plus de détails sur l'API, reportez-vous GetCoreDefinitionVersionà la section Référence des AWS CLI commandes.

`get-core-definition`

L'exemple de code suivant montre comment utiliser`get-core-definition`.

AWS CLI

Pour récupérer les détails d'une définition de base de Greengrass

L'`get-core-definition`exemple suivant permet de récupérer des informations sur la définition de base spécifiée. Pour récupérer les définitions IDs de base, utilisez la `list-core-definitions` commande.

```
aws greengrass get-core-definition \
```

```
    --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd",
    "CreationTimestamp": "2018-10-18T04:47:06.721Z",
    "Id": "237d6916-27cf-457f-ba0c-e86cfb5d25cd",
    "LastUpdatedTimestamp": "2018-10-18T04:47:06.721Z",
    "LatestVersion": "bd2cd6d4-2bc5-468a-8962-39e071e34b68",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd/versions/
bd2cd6d4-2bc5-468a-8962-39e071e34b68",
    "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous GetCoreDefinitionà la section Référence des AWS CLI commandes.

`get-deployment-status`

L'exemple de code suivant montre comment utiliser`get-deployment-status`.

AWS CLI

Pour récupérer le statut d'un déploiement

L'`get-deployment-status`exemple suivant récupère le statut du déploiement spécifié du groupe Greengrass spécifié. Pour obtenir l'ID de déploiement, utilisez la `list-deployments` commande et spécifiez l'ID de groupe.

```
aws greengrass get-deployment-status \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
    --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

Sortie :

```
{
    "DeploymentStatus": "Success",
    "DeploymentType": "NewDeployment",
```

```
    "UpdatedAt": "2019-06-18T17:04:44.761Z"
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeploymentStatusà la section Référence des AWS CLI commandes.

`get-device-definition-version`

L'exemple de code suivant montre comment utiliser`get-device-definition-version`.

AWS CLI

Pour obtenir une version de définition de l'appareil

L'`get-device-definition-version`exemple suivant extrait des informations sur la version spécifiée de la définition de périphérique spécifiée. Pour récupérer toutes les versions de la définition du périphérique, utilisez la `list-device-definition-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée à la définition de l'appareil, utilisez la `get-device-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-device-definition-version \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"
```

Sortie :

```
{
    "Definition": {
        "Devices": [
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02",
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
ExteriorTherm",
                "SyncShadow": true,
                "Id": "ExteriorTherm"
            },
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
InteriorTherm",
```

```
                "SyncShadow": true,
                "Id": "InteriorTherm"
            }
        ]
    },
    "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "CreationTimestamp": "2019-09-11T00:15:09.838Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeviceDefinitionVersionà la section Référence des AWS CLI commandes.

`get-device-definition`

L'exemple de code suivant montre comment utiliser`get-device-definition`.

AWS CLI

Pour obtenir une définition d'appareil

L'`get-device-definition`exemple suivant permet de récupérer des informations sur la définition de périphérique spécifiée. Pour récupérer les définitions IDs de votre appareil, utilisez la `list-device-definitions` commande.

```
aws greengrass get-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Sortie :

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "Name": "TemperatureSensors",
    "tags": {},
    "LastUpdatedTimestamp": "2019-09-11T00:19:03.698Z",
    "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
```

```
    "CreationTimestamp": "2019-09-11T00:11:06.197Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- Pour plus de détails sur l'API, reportez-vous GetDeviceDefinitionà la section Référence des AWS CLI commandes.

`get-function-definition-version`

L'exemple de code suivant montre comment utiliser`get-function-definition-version`.

AWS CLI

Pour récupérer des informations sur une version spécifique d'une fonction Lambda

Ce qui suit `get-function-definition-version` permet de récupérer des informations sur la version spécifiée de la définition de fonction spécifiée. Pour récupérer toutes les versions de la définition de fonction, utilisez la `list-function-definition-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée à la définition de la fonction, utilisez la `get-function-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-function-definition-version \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \
    --function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/9748fda7-1589-4fcc-ac94-
f5559e88678b",
    "CreationTimestamp": "2019-06-18T17:04:30.776Z",
    "Definition": {
        "Functions": [
            {
                "FunctionArn": "arn:aws:lambda:::function:GGIPDetector:1",
                "FunctionConfiguration": {
                    "Environment": {},
                    "MemorySize": 32768,
```

```
                    "Pinned": true,
                    "Timeout": 3
                },
                "Id": "26b69bdb-e547-46bc-9812-84ec04b6cc8c"
            },
            {
                "FunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",
                "FunctionConfiguration": {
                    "EncodingType": "json",
                    "Environment": {
                        "Variables": {}
                    },
                    "MemorySize": 16384,
                    "Pinned": true,
                    "Timeout": 25
                },
                "Id": "384465a8-eedf-48c6-b793-4c35f7bfae9b"
            }
        ]
    },
    "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
    "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
}
```

- Pour plus de détails sur l'API, reportez-vous GetFunctionDefinitionVersionà la section Référence des AWS CLI commandes.

`get-function-definition`

L'exemple de code suivant montre comment utiliser`get-function-definition`.

AWS CLI

Pour récupérer une définition de fonction

L'`get-function-definition`exemple suivant affiche les détails de la définition de fonction spécifiée. Pour récupérer les définitions IDs de vos fonctions, utilisez la `list-function-definitions` commande.

```
aws greengrass get-function-definition \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
    "CreationTimestamp": "2019-06-18T16:21:21.431Z",
    "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
    "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
    "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
    "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous GetFunctionDefinitionà la section Référence des AWS CLI commandes.

`get-group-certificate-authority`

L'exemple de code suivant montre comment utiliser`get-group-certificate-authority`.

AWS CLI

Pour récupérer l'autorité de certification associée à un groupe Greengrass

L'`get-group-certificate-authority`exemple suivant récupère l'autorité de certification (CA) associée au groupe Greengrass spécifié. Pour obtenir l'ID de l'autorité de certification, utilisez la `list-group-certificate-authorities` commande et spécifiez l'ID du groupe.

```
aws greengrass get-group-certificate-authority \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
    --certificate-authority-
id "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
```

Sortie :

```
{
    "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
```

```
    "GroupCertificateAuthorityId":
 "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
    "PemEncodedCertificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBWEXAMPLEGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDEXAMPLEEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAEXAMPLESBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jEXAMPLENMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0EXAMPLEBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWEXAMPLEDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWEXAMPLEgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5EXAMPLE8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CEXAMPLE93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswYEXAMPLEgpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKEXAMPLEAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n"
}
```

- Pour plus de détails sur l'API, reportez-vous GetGroupCertificateAuthorityà la section Référence des AWS CLI commandes.

`get-group-certificate-configuration`

L'exemple de code suivant montre comment utiliser`get-group-certificate-configuration`.

AWS CLI

Pour récupérer la configuration de l'autorité de certification utilisée par le groupe Greengrass

L'`get-group-certificate-configuration`exemple suivant récupère la configuration de l'autorité de certification (CA) utilisée par le groupe Greengrass spécifié.

```
aws greengrass get-group-certificate-configuration \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Sortie :

```
{
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
```

```
    "CertificateExpiryInMilliseconds": 604800000,
    "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"
}
```

- Pour plus de détails sur l'API, reportez-vous GetGroupCertificateConfigurationà la section Référence des AWS CLI commandes.

`get-group-version`

L'exemple de code suivant montre comment utiliser`get-group-version`.

AWS CLI

Pour récupérer des informations sur une version d'un groupe Greengrass

L'`get-group-version`exemple suivant extrait des informations sur la version spécifiée du groupe spécifié. Pour récupérer toutes les versions du groupe, utilisez la `list-group-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée au groupe, utilisez la `get-group` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-group-version \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"  \
    --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
    "CreationTimestamp": "2019-06-18T17:04:30.915Z",
    "Definition": {
        "CoreDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-
fd9d-4312-a8fd-ffa9404a20e0",
        "FunctionDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
        "SubscriptionDefinitionVersionArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/definition/subscriptions/70e49321-83d5-45d2-
bc09-81f4917ae152/versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

```
    },
    "Id": "1013db12-8b58-45ff-acc7-704248f66731",
    "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
}
```

- Pour plus de détails sur l'API, reportez-vous GetGroupVersionà la section Référence des AWS CLI commandes.

`get-group`

L'exemple de code suivant montre comment utiliser`get-group`.

AWS CLI

Pour récupérer des informations sur un groupe Greengrass

L'`get-group`exemple suivant permet de récupérer des informations sur le groupe Greengrass spécifié. Pour récupérer vos groupes, utilisez la `list-groups` commande. IDs

```
aws greengrass get-group \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731",
    "CreationTimestamp": "2019-06-18T16:21:21.457Z",
    "Id": "1013db12-8b58-45ff-acc7-704248f66731",
    "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",
    "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
    "Name": "GGGroup4Pi3",
    "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous GetGroupà la section Référence des AWS CLI commandes.

`get-logger-definition-version`

L'exemple de code suivant montre comment utiliser`get-logger-definition-version`.

AWS CLI

Pour récupérer des informations sur une version d'une définition d'enregistreur

L'`get-logger-definition-version`exemple suivant extrait des informations sur la version spécifiée de la définition d'enregistreur spécifiée. Pour récupérer toutes les versions IDs de la définition de l'enregistreur, utilisez la `list-logger-definition-versions` commande. Pour récupérer l'ID de la dernière version ajoutée à la définition de l'enregistreur, utilisez la `get-logger-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-logger-definition-version \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \
    --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
    "CreationTimestamp": "2019-05-08T16:10:13.866Z",
    "Definition": {
        "Loggers": []
    },
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
}
```

- Pour plus de détails sur l'API, reportez-vous GetLoggerDefinitionVersionà la section Référence des AWS CLI commandes.

`get-logger-definition`

L'exemple de code suivant montre comment utiliser`get-logger-definition`.

AWS CLI

Pour récupérer des informations sur la définition d'un enregistreur

L'`get-logger-definition`exemple suivant récupère les informations relatives à la définition de l'enregistreur spécifiée. Pour récupérer les définitions IDs de votre enregistreur, utilisez la `list-logger-definitions` commande.

```
aws greengrass get-logger-definition \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "CreationTimestamp": "2019-05-08T16:10:13.809Z",
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
    "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
    "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous GetLoggerDefinitionà la section Référence des AWS CLI commandes.

`get-resource-definition-version`

L'exemple de code suivant montre comment utiliser`get-resource-definition-version`.

AWS CLI

Pour récupérer des informations sur une version spécifique d'une définition de ressource

L'`get-resource-definition-version`exemple suivant extrait des informations sur la version spécifiée de la définition de ressource spécifiée. Pour récupérer toutes les versions de la définition de ressource, utilisez la `list-resource-definition-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée à la définition de ressource, utilisez la `get-resource-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-resource-definition-version \
```

```
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
    --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
    "CreationTimestamp": "2019-06-19T16:40:59.392Z",
    "Definition": {
        "Resources": [
            {
                "Id": "26ff3f7b-839a-4217-9fdc-a218308b3963",
                "Name": "usb-port",
                "ResourceDataContainer": {
                    "LocalDeviceResourceData": {
                        "GroupOwnerSetting": {
                            "AutoAddGroupOwner": false
                        },
                        "SourcePath": "/dev/bus/usb"
                    }
                }
            }
        ]
    },
    "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourceDefinitionVersionà la section Référence des AWS CLI commandes.

`get-resource-definition`

L'exemple de code suivant montre comment utiliser`get-resource-definition`.

AWS CLI

Pour récupérer des informations sur une définition de ressource

L'`get-resource-definition`exemple suivant permet de récupérer des informations sur la définition de ressource spécifiée. Pour récupérer les définitions IDs de vos ressources, utilisez la `list-resource-definitions` commande.

```
aws greengrass get-resource-definition \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "CreationTimestamp": "2019-06-19T16:40:59.261Z",
    "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
    "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
    "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourceDefinitionà la section Référence des AWS CLI commandes.

`get-service-role-for-account`

L'exemple de code suivant montre comment utiliser`get-service-role-for-account`.

AWS CLI

Pour récupérer les informations relatives au rôle de service associé à votre compte

L'`get-service-role-for-account`exemple suivant permet de récupérer des informations sur le rôle de service associé à votre AWS compte.

```
aws greengrass get-service-role-for-account
```

Sortie :

```
{
```

```
    "AssociatedAt": "2018-10-18T15:59:20Z",
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Pour plus d'informations, consultez Greengrass Service Role dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous GetServiceRoleForAccountà la section Référence des AWS CLI commandes.

`get-subscription-definition-version`

L'exemple de code suivant montre comment utiliser`get-subscription-definition-version`.

AWS CLI

Pour récupérer des informations sur une version spécifique d'une définition d'abonnement

L'`get-subscription-definition-version`exemple suivant extrait des informations sur la version spécifiée de la définition d'abonnement spécifiée. Pour récupérer toutes les versions de la définition d'abonnement, utilisez la `list-subscription-definition-versions` commande. IDs Pour récupérer l'ID de la dernière version ajoutée à la définition de l'abonnement, utilisez la `get-subscription-definition` commande et vérifiez la `LatestVersion` propriété.

```
aws greengrass get-subscription-definition-version \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \
    --subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/88ae8699-12ac-4663-
ba3f-4d7f0519140b",
    "CreationTimestamp": "2019-06-18T17:03:52.499Z",
    "Definition": {
        "Subscriptions": [
            {
                "Id": "692c4484-d89f-4f64-8edd-1a041a65e5b6",
                "Source": "arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",
```

```
                "Subject": "hello/world",
                "Target": "cloud"
            }
        ]
    },
    "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
    "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
}
```

- Pour plus de détails sur l'API, reportez-vous GetSubscriptionDefinitionVersionà la section Référence des AWS CLI commandes.

`get-subscription-definition`

L'exemple de code suivant montre comment utiliser`get-subscription-definition`.

AWS CLI

Pour récupérer des informations sur la définition d'un abonnement

L'`get-subscription-definition`exemple suivant permet de récupérer des informations sur la définition d'abonnement spécifiée. Pour récupérer les définitions IDs de votre abonnement, utilisez la `list-subscription-definitions` commande.

```
aws greengrass get-subscription-definition \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

Sortie :

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
    "CreationTimestamp": "2019-06-18T17:03:52.392Z",
    "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
    "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
    "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",
    "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous GetSubscriptionDefinitionà la section Référence des AWS CLI commandes.

`get-thing-runtime-configuration`

L'exemple de code suivant montre comment utiliser`get-thing-runtime-configuration`.

AWS CLI

Pour récupérer la configuration d'exécution d'un noyau Greengrass

L'`get-thing-runtime-configuration`exemple suivant récupère la configuration d'exécution d'un noyau Greengrass. Avant de pouvoir récupérer la configuration d'exécution, vous devez utiliser la `update-thing-runtime-configuration` commande pour créer une configuration d'exécution pour le noyau.

```
aws greengrass get-thing-runtime-configuration \
    --thing-name SampleGreengrassCore
```

Sortie :

```
{
    "RuntimeConfiguration": {
        "TelemetryConfiguration": {
            "ConfigurationSyncStatus": "OutOfSync",
            "Telemetry": "On"
        }
    }
}
```

Pour plus d'informations, consultez la section Configuration des paramètres de télémétrie dans le manuel IoT AWS Greengrass Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetThingRuntimeConfigurationà la section Référence des AWS CLI commandes.

`list-bulk-deployment-detailed-reports`

L'exemple de code suivant montre comment utiliser`list-bulk-deployment-detailed-reports`.

AWS CLI

Pour répertorier les informations relatives aux déploiements individuels dans le cadre d'un déploiement en masse

L'`list-bulk-deployment-detailed-reports`exemple suivant affiche des informations sur les déploiements individuels dans le cadre d'une opération de déploiement en masse, y compris le statut.

```
aws greengrass list-bulk-deployment-detailed-reports \
    --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

Sortie :

```
{
    "Deployments": [
        {
            "DeploymentType": "NewDeployment",
            "DeploymentStatus": "Success",
            "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/
deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/
versions/123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "CreatedAt": "2020-01-21T21:34:16.501Z"
        },
        {
            "DeploymentType": "NewDeployment",
            "DeploymentStatus": "InProgress",
            "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/
deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/versions/a1b2c3d4-5678-90ab-cdef-
EXAMPLE66666",
            "CreatedAt": "2020-01-21T21:34:16.486Z"
        },
        ...
    ]
```

```
}
```

Pour plus d'informations, consultez la section Créer des déploiements groupés pour les groupes dans le Guide du AWS développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous ListBulkDeploymentDetailedReportsà la section Référence des AWS CLI commandes.

`list-bulk-deployments`

L'exemple de code suivant montre comment utiliser`list-bulk-deployments`.

AWS CLI

Pour répertorier les déploiements groupés

L'`list-bulk-deployments`exemple suivant répertorie tous les déploiements en masse.

```
aws greengrass list-bulk-deployments
```

Sortie :

```
{
    "BulkDeployments": [
        {
            "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
            "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
            "CreatedAt": "2019-06-25T16:11:33.265Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Créer des déploiements groupés pour les groupes dans le Guide du AWS développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous ListBulkDeploymentsà la section Référence des AWS CLI commandes.

`list-connector-definition-versions`

L'exemple de code suivant montre comment utiliser`list-connector-definition-versions`.

AWS CLI

Pour répertorier les versions disponibles pour une définition de connecteur

L'`list-connector-definition-versions`exemple suivant répertorie les versions disponibles pour la définition de connecteur spécifiée. Utilisez la `list-connector-definitions` commande pour obtenir l'ID de définition du connecteur.

```
aws greengrass list-connector-definition-versions \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Sortie :

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
            "CreationTimestamp": "2019-06-19T19:30:01.300Z",
            "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
            "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
        }
    ]
}
```

Pour plus d'informations, consultez la section Intégrer aux services et aux protocoles à l'aide des connecteurs Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous ListConnectorDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-connector-definitions`

L'exemple de code suivant montre comment utiliser`list-connector-definitions`.

AWS CLI

Pour répertorier les connecteurs Greengrass définis

L'`list-connector-definitions`exemple suivant répertorie tous les connecteurs Greengrass définis pour votre AWS compte.

```
aws greengrass list-connector-definitions
```

Sortie :

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
            "CreationTimestamp": "2019-06-19T19:30:01.300Z",
            "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
            "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
            "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/
versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",
            "Name": "MySNSConnector"
        }
    ]
}
```

Pour plus d'informations, consultez la section Intégrer aux services et aux protocoles à l'aide des connecteurs Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous ListConnectorDefinitionsà la section Référence des AWS CLI commandes.

`list-core-definition-versions`

L'exemple de code suivant montre comment utiliser`list-core-definition-versions`.

AWS CLI

Pour répertorier les versions d'une définition de base de Greengrass

L'`list-core-definitions`exemple suivant répertorie toutes les versions de la définition de base Greengrass spécifiée. Vous pouvez utiliser la `list-core-definitions` commande pour obtenir l'ID de version.

```
aws greengrass list-core-definition-versions \
    --core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"
```

Sortie :

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-c5da-440c-
a97b-084e62593b4c",
            "CreationTimestamp": "2019-06-18T16:14:17.709Z",
            "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
            "Version": "467c36e4-c5da-440c-a97b-084e62593b4c"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListCoreDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-core-definitions`

L'exemple de code suivant montre comment utiliser`list-core-definitions`.

AWS CLI

Pour répertorier les définitions de base de Greengrass

L'`list-core-definitions`exemple suivant répertorie toutes les définitions de base de Greengrass pour votre AWS compte.

```
aws greengrass list-core-definitions
```

Sortie :

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d",
            "CreationTimestamp": "2018-10-17T04:30:32.786Z",
            "Id": "0507843c-c1ef-4f06-b051-817030df7e7d",
            "LastUpdatedTimestamp": "2018-10-17T04:30:32.786Z",
```

```
            "LatestVersion": "bcdf9e86-3793-491e-93af-3cdfbf4e22b7",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d/versions/
bcdf9e86-3793-491e-93af-3cdfbf4e22b7"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/31c22500-3509-4271-bafd-cf0655cda438",
            "CreationTimestamp": "2019-06-18T16:24:16.064Z",
            "Id": "31c22500-3509-4271-bafd-cf0655cda438",
            "LastUpdatedTimestamp": "2019-06-18T16:24:16.064Z",
            "LatestVersion": "2f350395-6d09-4c8a-8336-9ae5b57ace84",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/31c22500-3509-4271-bafd-cf0655cda438/
versions/2f350395-6d09-4c8a-8336-9ae5b57ace84"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46",
            "CreationTimestamp": "2019-06-18T16:21:21.351Z",
            "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
            "LastUpdatedTimestamp": "2019-06-18T16:21:21.351Z",
            "LatestVersion": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-
fd9d-4312-a8fd-ffa9404a20e0"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7",
            "CreationTimestamp": "2019-06-18T16:14:17.709Z",
            "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
            "LastUpdatedTimestamp": "2019-06-18T16:14:17.709Z",
            "LatestVersion": "467c36e4-c5da-440c-a97b-084e62593b4c",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-
c5da-440c-a97b-084e62593b4c"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListCoreDefinitionsà la section Référence des AWS CLI commandes.

`list-deployments`

L'exemple de code suivant montre comment utiliser`list-deployments`.

AWS CLI

Pour répertorier les déploiements d'un groupe Greengrass

L'`list-deployments`exemple suivant répertorie les déploiements pour le groupe Greengrass spécifié. Vous pouvez utiliser la `list-groups` commande pour rechercher l'identifiant de votre groupe.

```
aws greengrass list-deployments \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Sortie :

```
{
    "Deployments": [
        {
            "CreatedAt": "2019-06-18T17:04:32.702Z",
            "DeploymentId": "1065b8a0-812b-4f21-9d5d-e89b232a530f",
            "DeploymentType": "NewDeployment",
            "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeploymentsà la section Référence des AWS CLI commandes.

`list-device-definition-versions`

L'exemple de code suivant montre comment utiliser`list-device-definition-versions`.

AWS CLI

Pour répertorier les versions d'une définition d'appareil

L'`list-device-definition-versions`exemple suivant affiche les versions de définition de périphérique associées à la définition de périphérique spécifiée.

```
aws greengrass list-device-definition-versions \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Sortie :

```
{
    "Versions": [
        {
            "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
            "CreationTimestamp": "2019-09-11T00:15:09.838Z",
            "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
        },
        {
            "Version": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
            "CreationTimestamp": "2019-09-11T00:11:06.197Z",
            "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeviceDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-device-definitions`

L'exemple de code suivant montre comment utiliser`list-device-definitions`.

AWS CLI

Pour répertorier les définitions de vos appareils

L'`list-device-definitions`exemple suivant affiche des détails sur les définitions d'appareils de votre AWS compte dans la AWS région spécifiée.

```
aws greengrass list-device-definitions \
    --region us-west-2
```

Sortie :

```
{
    "Definitions": [
        {
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab/versions/
c777b0f5-1059-449b-beaa-f003ebc56c34",
            "LastUpdatedTimestamp": "2019-06-14T15:42:09.059Z",
            "LatestVersion": "c777b0f5-1059-449b-beaa-f003ebc56c34",
            "CreationTimestamp": "2019-06-14T15:42:09.059Z",
            "Id": "50f3274c-3f0a-4f57-b114-6f46085281ab",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab"
        },
        {
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/e01951c9-6134-479a-969a-1a15cac11c40/
versions/514d57aa-4ee6-401c-9fac-938a9f7a51e5",
            "Name": "TestDeviceDefinition",
            "LastUpdatedTimestamp": "2019-04-16T23:17:43.245Z",
            "LatestVersion": "514d57aa-4ee6-401c-9fac-938a9f7a51e5",
            "CreationTimestamp": "2019-04-16T23:17:43.245Z",
            "Id": "e01951c9-6134-479a-969a-1a15cac11c40",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/e01951c9-6134-479a-969a-1a15cac11c40"
        },
        {
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
            "Name": "TemperatureSensors",
            "LastUpdatedTimestamp": "2019-09-10T00:19:03.698Z",
            "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
            "CreationTimestamp": "2019-09-11T00:11:06.197Z",
            "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
        }
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListDeviceDefinitionsà la section Référence des AWS CLI commandes.

`list-function-definition-versions`

L'exemple de code suivant montre comment utiliser`list-function-definition-versions`.

AWS CLI

Pour répertorier les versions d'une fonction Lambda

L'`list-function-definition-versions`exemple suivant répertorie toutes les versions de la fonction Lambda spécifiée. Vous pouvez utiliser la `list-function-definitions` commande pour obtenir l'identifiant.

```
aws greengrass list-function-definition-versions \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

Sortie :

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
            "CreationTimestamp": "2019-06-18T17:04:30.776Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9b08df77-26f2-4c29-93d2-769715edcfec",
            "CreationTimestamp": "2019-06-18T17:02:44.087Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "9b08df77-26f2-4c29-93d2-769715edcfec"
        },
        {
```

```
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/4236239f-94f7-4b90-a2f8-2a24c829d21e",
            "CreationTimestamp": "2019-06-18T17:01:42.284Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "4236239f-94f7-4b90-a2f8-2a24c829d21e"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/343408bb-549a-4fbe-b043-853643179a39",
            "CreationTimestamp": "2019-06-18T16:21:21.431Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "343408bb-549a-4fbe-b043-853643179a39"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListFunctionDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-function-definitions`

L'exemple de code suivant montre comment utiliser`list-function-definitions`.

AWS CLI

Pour répertorier les fonctions Lambda

L'`list-function-definitions`exemple suivant répertorie toutes les fonctions Lambda définies pour votre AWS compte.

```
aws greengrass list-function-definitions
```

Sortie :

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960",
            "CreationTimestamp": "2018-10-17T04:30:32.884Z",
```

```
            "Id": "017970a5-8952-46dd-b1c1-020b3ae8e960",
            "LastUpdatedTimestamp": "2018-10-17T04:30:32.884Z",
            "LatestVersion": "4380b302-790d-4ed8-92bf-02e88afecb15",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960/
versions/4380b302-790d-4ed8-92bf-02e88afecb15"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "CreationTimestamp": "2019-06-18T16:21:21.431Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
            "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/6598e653-a262-440c-9967-e2697f64da7b",
            "CreationTimestamp": "2019-06-18T16:24:16.123Z",
            "Id": "6598e653-a262-440c-9967-e2697f64da7b",
            "LastUpdatedTimestamp": "2019-06-18T16:24:16.123Z",
            "LatestVersion": "38bc6ccd-98a2-4ce7-997e-16c84748fae4",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/6598e653-a262-440c-9967-e2697f64da7b/
versions/38bc6ccd-98a2-4ce7-997e-16c84748fae4"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/c668df84-fad2-491b-95f4-655d2cad7885",
            "CreationTimestamp": "2019-06-18T16:14:17.784Z",
            "Id": "c668df84-fad2-491b-95f4-655d2cad7885",
            "LastUpdatedTimestamp": "2019-06-18T16:14:17.784Z",
            "LatestVersion": "37dd68c4-a64f-40ba-aa13-71fecc3ebded",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/c668df84-fad2-491b-95f4-655d2cad7885/
versions/37dd68c4-a64f-40ba-aa13-71fecc3ebded"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListFunctionDefinitionsà la section Référence des AWS CLI commandes.

`list-group-certificate-authorities`

L'exemple de code suivant montre comment utiliser`list-group-certificate-authorities`.

AWS CLI

Pour répertorier le courant CAs d'un groupe

L'`list-group-certificate-authorities`exemple suivant répertorie les autorités de certification actuelles (CAs) pour le groupe Greengrass spécifié.

```
aws greengrass list-group-certificate-authorities \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Sortie :

```
{
    "GroupCertificateAuthorities": [
        {
            "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
            "GroupCertificateAuthorityId":
 "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGroupCertificateAuthoritiesà la section Référence des AWS CLI commandes.

`list-group-versions`

L'exemple de code suivant montre comment utiliser`list-group-versions`.

AWS CLI

Pour répertorier les versions d'un groupe Greengrass

L'`list-group-versions`exemple suivant répertorie les versions du groupe Greengrass spécifié.

```
aws greengrass list-group-versions \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Sortie :

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
            "CreationTimestamp": "2019-06-18T17:04:30.915Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/4340669d-
d14d-44e3-920c-46c928750750",
            "CreationTimestamp": "2019-06-18T17:03:52.663Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "4340669d-d14d-44e3-920c-46c928750750"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
versions/1b06e099-2d5b-4f10-91b9-78c4e060f5da",
            "CreationTimestamp": "2019-06-18T17:02:44.189Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "1b06e099-2d5b-4f10-91b9-78c4e060f5da"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/2d3f27f1-3b43-4554-
ab7a-73ec30477efe",
            "CreationTimestamp": "2019-06-18T17:01:42.401Z",
```

```
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "2d3f27f1-3b43-4554-ab7a-73ec30477efe"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/d20f7ae9-3444-4c1c-b025-
e2ede23cdd31",
            "CreationTimestamp": "2019-06-18T16:21:21.457Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "d20f7ae9-3444-4c1c-b025-e2ede23cdd31"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGroupVersionsà la section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour répertorier les groupes Greengrass

L'`list-groups`exemple suivant répertorie tous les groupes Greengrass définis dans votre AWS compte.

```
aws greengrass list-groups
```

Sortie :

```
{
    "Groups": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731",
            "CreationTimestamp": "2019-06-18T16:21:21.457Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",
            "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",
```

```
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
            "Name": "GGGroup4Pi3"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
            "CreationTimestamp": "2018-10-31T21:52:46.603Z",
            "Id": "1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
            "LastUpdatedTimestamp": "2018-10-31T21:52:46.603Z",
            "LatestVersion": "749af901-60ab-456f-a096-91b12d983c29",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/versions/749af901-60ab-456f-
a096-91b12d983c29",
            "Name": "MyTestGroup"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/504b5c8d-bbed-4635-aff1-48ec5b586db5",
            "CreationTimestamp": "2018-12-31T21:39:36.771Z",
            "Id": "504b5c8d-bbed-4635-aff1-48ec5b586db5",
            "LastUpdatedTimestamp": "2018-12-31T21:39:36.771Z",
            "LatestVersion": "46911e8e-f9bc-4898-8b63-59c7653636ec",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/504b5c8d-bbed-4635-aff1-48ec5b586db5/versions/46911e8e-
f9bc-4898-8b63-59c7653636ec",
            "Name": "smp-ggrass-group"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGroupsà la section Référence des AWS CLI commandes.

`list-logger-definition-versions`

L'exemple de code suivant montre comment utiliser`list-logger-definition-versions`.

AWS CLI

Pour obtenir la liste des versions d'une définition d'enregistreur

L'`list-logger-definition-versions`exemple suivant permet d'obtenir une liste de toutes les versions de la définition d'enregistreur spécifiée.

```
aws greengrass list-logger-definition-versions \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

Sortie :

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
            "CreationTimestamp": "2019-05-08T16:10:13.866Z",
            "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/3ec6d3af-eb85-48f9-
a16d-1c795fe696d7",
            "CreationTimestamp": "2019-05-08T16:10:13.809Z",
            "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "Version": "3ec6d3af-eb85-48f9-a16d-1c795fe696d7"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListLoggerDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-logger-definitions`

L'exemple de code suivant montre comment utiliser`list-logger-definitions`.

AWS CLI

Pour obtenir une liste des définitions des enregistreurs

L'`list-logger-definitions`exemple suivant répertorie toutes les définitions d'enregistreurs pour votre AWS compte.

```
aws greengrass list-logger-definitions
```

Sortie :

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "CreationTimestamp": "2019-05-08T16:10:13.809Z",
            "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
            "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/
versions/5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListLoggerDefinitionsà la section Référence des AWS CLI commandes.

`list-resource-definition-versions`

L'exemple de code suivant montre comment utiliser`list-resource-definition-versions`.

AWS CLI

Pour répertorier les versions d'une définition de ressource

L'`list-resource-definition-versions`exemple suivant répertorie les versions de la ressource Greengrass spécifiée.

```
aws greengrass list-resource-definition-versions \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Sortie :

```
{
    "Versions": [
```

```
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
            "CreationTimestamp": "2019-06-19T16:40:59.392Z",
            "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/432d92f6-12de-4ec9-a704-619a942a62aa",
            "CreationTimestamp": "2019-06-19T16:40:59.261Z",
            "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "Version": "432d92f6-12de-4ec9-a704-619a942a62aa"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-resource-definitions`

L'exemple de code suivant montre comment utiliser`list-resource-definitions`.

AWS CLI

Pour répertorier les ressources définies

L'`list-resource-definitions`exemple suivant répertorie les ressources définies pour AWS IoT Greengrass à utiliser.

```
aws greengrass list-resource-definitions
```

Sortie :

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
```

```
            "CreationTimestamp": "2019-06-19T16:40:59.261Z",
            "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
            "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
            "CreationTimestamp": "2019-06-19T21:51:28.212Z",
            "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
            "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
            "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
            "Name": "MyGreengrassResources"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceDefinitionsà la section Référence des AWS CLI commandes.

`list-subscription-definition-versions`

L'exemple de code suivant montre comment utiliser`list-subscription-definition-versions`.

AWS CLI

Pour répertorier les versions d'une définition d'abonnement

L'`list-subscription-definition-versions`exemple suivant répertorie toutes les versions de l'abonnement spécifié. Vous pouvez utiliser la `list-subscription-definitions` commande pour rechercher l'identifiant de l'abonnement.

```
aws greengrass list-subscription-definition-versions \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

Sortie :

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",
            "CreationTimestamp": "2019-06-18T17:03:52.499Z",
            "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
            "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/7e320ba3-
c369-4069-a2f0-90acb7f219d6",
            "CreationTimestamp": "2019-06-18T17:03:52.392Z",
            "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
            "Version": "7e320ba3-c369-4069-a2f0-90acb7f219d6"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSubscriptionDefinitionVersionsà la section Référence des AWS CLI commandes.

`list-subscription-definitions`

L'exemple de code suivant montre comment utiliser`list-subscription-definitions`.

AWS CLI

Pour obtenir les définitions des abonnements à une liste

L'`list-subscription-definitions`exemple suivant répertorie tous les abonnements AWS IoT Greengrass définis dans votre AWS compte.

```
aws greengrass list-subscription-definitions
```

Sortie :

```
{
```

```
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
            "CreationTimestamp": "2019-06-18T17:03:52.392Z",
            "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
            "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
            "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967",
            "CreationTimestamp": "2018-10-18T15:45:34.024Z",
            "Id": "cd6f1c37-d9a4-4e90-be94-01a7404f5967",
            "LastUpdatedTimestamp": "2018-10-18T15:45:34.024Z",
            "LatestVersion": "d1cf8fac-284f-4f6a-98fe-a2d36d089373",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967/versions/
d1cf8fac-284f-4f6a-98fe-a2d36d089373"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b",
            "CreationTimestamp": "2018-10-22T17:09:31.429Z",
            "Id": "fa81bc84-3f59-4377-a84b-5d0134da359b",
            "LastUpdatedTimestamp": "2018-10-22T17:09:31.429Z",
            "LatestVersion": "086d1b08-b25a-477c-a16f-6f9b3a9c295a",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b/
versions/086d1b08-b25a-477c-a16f-6f9b3a9c295a"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSubscriptionDefinitionsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises associées à une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises et leurs valeurs associées à la ressource spécifiée.

```
aws greengrass list-tags-for-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Sortie :

```
{
    "tags": {
        "ResourceSubType": "USB",
        "ResourceType": "Device"
    }
}
```

Pour plus d'informations, consultez la section Marquage de vos ressources Greengrass dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`reset-deployments`

L'exemple de code suivant montre comment utiliser`reset-deployments`.

AWS CLI

Pour nettoyer les informations de déploiement d'un groupe Greengrass

L'`reset-deployments`exemple suivant nettoie les informations de déploiement pour le groupe Greengrass spécifié. Lorsque vous ajoutez le`--force option`, les informations de déploiement sont réinitialisées sans attendre que le périphérique principal réponde.

```
aws greengrass reset-deployments \
    --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
    --force
```

Sortie :

```
{
    "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/
deployments/7dd4e356-9882-46a3-9e28-6d21900c011a",
    "DeploymentId": "7dd4e356-9882-46a3-9e28-6d21900c011a"
}
```

Pour plus d'informations, consultez la section Réinitialiser les déploiements dans le guide du AWS développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous ResetDeploymentsà la section Référence des AWS CLI commandes.

`start-bulk-deployment`

L'exemple de code suivant montre comment utiliser`start-bulk-deployment`.

AWS CLI

Pour démarrer une opération de déploiement en masse

L'`start-bulk-deployment`exemple suivant lance une opération de déploiement en masse, en utilisant un fichier stocké dans un compartiment S3 pour spécifier les groupes à déployer.

```
aws greengrass start-bulk-deployment \
    --cli-input-json "{\"InputFileUri\":\"https://gg-group-deployment1.s3-us-
west-2.amazonaws.com/MyBulkDeploymentInputFile.txt\", \"ExecutionRoleArn\":
\"arn:aws:iam::123456789012:role/ggCreateDeploymentRole\",\"AmznClientToken\":
\"yourAmazonClientToken\"}"
```

Sortie :

```
{
    "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
    "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
}
```

Pour plus d'informations, consultez la section Créer des déploiements groupés pour les groupes dans le Guide du AWS développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous StartBulkDeploymentà la section Référence des AWS CLI commandes.

`stop-bulk-deployment`

L'exemple de code suivant montre comment utiliser`stop-bulk-deployment`.

AWS CLI

Pour arrêter un déploiement en masse

L'`stop-bulk-deployment`exemple suivant arrête le déploiement en bloc spécifié. Si vous essayez d'arrêter un déploiement groupé qui est terminé, vous recevez un message d'erreur : `InvalidInputException: Cannot change state of finished execution.`

```
aws greengrass stop-bulk-deployment \
    --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer des déploiements groupés pour les groupes dans le Guide du AWS développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous StopBulkDeploymentà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour appliquer des balises à une ressource

L'`tag-resource`exemple suivant applique deux balises, `ResourceType` et`ResourceSubType`, à la ressource Greengrass spécifiée. Cette opération peut à la fois ajouter de nouvelles balises et valeurs ou mettre à jour la valeur des balises existantes. Utilisez la `untag-resource` commande pour supprimer un tag.

```
aws greengrass tag-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
```

```
    --tags "ResourceType=Device,ResourceSubType=USB"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources Greengrass dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une balise et sa valeur d'une ressource

L'`untag-resource`exemple suivant supprime la balise dont la clé `Category` appartient au groupe Greengrass spécifié. Si la clé `Category` n'existe pas pour la ressource spécifiée, aucune erreur n'est renvoyée.

```
aws greengrass untag-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731" \
    --tag-keys "Category"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources Greengrass dans le guide du développeur AWS IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-connectivity-info`

L'exemple de code suivant montre comment utiliser`update-connectivity-info`.

AWS CLI

Pour mettre à jour les informations de connectivité d'un noyau Greengrass

L'`update-connectivity-info`exemple suivant modifie les points de terminaison que les appareils peuvent utiliser pour se connecter au cœur Greengrass spécifié. Les informations de connectivité sont une liste d'adresses IP ou de noms de domaine, avec les numéros de port correspondants et les métadonnées facultatives définies par le client. Il se peut que vous deviez mettre à jour les informations de connectivité lorsque le réseau local change.

```
aws greengrass update-connectivity-info \
    --thing-name "MyGroup_Core" \
    --connectivity-info "[{\"Metadata\":\"\",\"PortNumber\":8883,\"HostAddress\":
\"127.0.0.1\",\"Id\":\"localhost_127.0.0.1_0\"},{\"Metadata\":\"\",\"PortNumber
\":8883,\"HostAddress\":\"192.168.1.3\",\"Id\":\"localIP_192.168.1.3\"}]"
```

Sortie :

```
{
    "Version": "312de337-59af-4cf9-a278-2a23bd39c300"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateConnectivityInfoà la section Référence des AWS CLI commandes.

`update-connector-definition`

L'exemple de code suivant montre comment utiliser`update-connector-definition`.

AWS CLI

Pour mettre à jour le nom d'une définition de connecteur

L'`update-connector-definition`exemple suivant met à jour le nom de la définition de connecteur spécifiée. Si vous souhaitez mettre à jour les détails du connecteur, utilisez la `create-connector-definition-version` commande pour créer une nouvelle version.

```
aws greengrass update-connector-definition \
    --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
    --name "GreengrassConnectors2019"
```

Pour plus d'informations, consultez la section Intégrer aux services et aux protocoles à l'aide de connecteurs dans le AWS guide du développeur IoT Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateConnectorDefinitionà la section Référence des AWS CLI commandes.

`update-core-definition`

L'exemple de code suivant montre comment utiliser`update-core-definition`.

AWS CLI

Pour mettre à jour une définition de base

L'`update-core-definition`exemple suivant modifie le nom de la définition de base spécifiée. Vous ne pouvez mettre à jour que la `name` propriété d'une définition de base.

```
aws greengrass update-core-definition \
    --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
    --name "MyCoreDevices"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Configurer le cœur AWS IoT Greengrass dans le Guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateCoreDefinitionà la section Référence des AWS CLI commandes.

`update-device-definition`

L'exemple de code suivant montre comment utiliser`update-device-definition`.

AWS CLI

Pour mettre à jour la définition d'un appareil

L'`update-device-definition`exemple suivant modifie le nom de la définition de périphérique spécifiée. Vous pouvez uniquement mettre à jour la `name` propriété d'une définition d'appareil.

```
aws greengrass update-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --name "TemperatureSensors"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateDeviceDefinitionà la section Référence des AWS CLI commandes.

`update-function-definition`

L'exemple de code suivant montre comment utiliser`update-function-definition`.

AWS CLI

Pour mettre à jour le nom d'une définition de fonction

L'`update-function-definition`exemple suivant met à jour le nom de la définition de fonction spécifiée. Si vous souhaitez mettre à jour les détails de la fonction, utilisez la `create-function-definition-version` commande pour créer une nouvelle version.

```
aws greengrass update-function-definition \
    --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \
    --name ObsoleteFunction
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Exécuter des fonctions Lambda locales dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateFunctionDefinitionà la section Référence des AWS CLI commandes.

`update-group-certificate-configuration`

L'exemple de code suivant montre comment utiliser`update-group-certificate-configuration`.

AWS CLI

Pour mettre à jour l'expiration des certificats d'un groupe

L'`update-group-certificate-configuration`exemple suivant définit une expiration de 10 jours pour les certificats générés pour le groupe spécifié.

```
aws greengrass update-group-certificate-configuration \
    --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \
```

```
    --certificate-expiry-in-milliseconds 864000000
```

Sortie :

```
{
    "CertificateExpiryInMilliseconds": 864000000,
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
    "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
}
```

Pour plus d'informations, consultez AWS IoT Greengrass Security dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupCertificateConfigurationà la section Référence des AWS CLI commandes.

`update-group`

L'exemple de code suivant montre comment utiliser`update-group`.

AWS CLI

Pour mettre à jour le nom du groupe

L'`update-group`exemple suivant met à jour le nom du groupe Greengrass spécifié. Si vous souhaitez mettre à jour les informations du groupe, utilisez la `create-group-version` commande pour créer une nouvelle version.

```
aws greengrass update-group \
    --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
    --name TestGroup4of6
```

Pour plus d'informations, voir Configurer AWS IoT Greengrass on IoT dans le Guide du développeur AWS IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupà la section Référence des AWS CLI commandes.

`update-logger-definition`

L'exemple de code suivant montre comment utiliser`update-logger-definition`.

AWS CLI

Pour mettre à jour la définition d'un enregistreur

L'`update-logger-definition`exemple suivant modifie le nom de la définition d'enregistreur spécifiée. Vous pouvez uniquement mettre à jour la `name` propriété d'une définition d'enregistreur.

```
aws greengrass update-logger-definition \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
    --name "LoggingConfigsForSensors"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance à AWS l'aide des journaux IoT Greengrass dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateLoggerDefinitionà la section Référence des AWS CLI commandes.

`update-resource-definition`

L'exemple de code suivant montre comment utiliser`update-resource-definition`.

AWS CLI

Pour mettre à jour le nom d'une définition de ressource

L'`update-resource-definition`exemple suivant met à jour le nom de la définition de ressource spécifiée. Si vous souhaitez modifier les détails de la ressource, utilisez la `create-resource-definition-version` commande pour créer une nouvelle version.

```
aws greengrass update-resource-definition \
    --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \
    --name GreengrassConnectorResources
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Accès aux ressources locales à l'aide de fonctions et de connecteurs Lambda dans le guide du développeur IoT AWS Greengrass.

- Pour plus de détails sur l'API, reportez-vous UpdateResourceDefinitionà la section Référence des AWS CLI commandes.

`update-subscription-definition`

L'exemple de code suivant montre comment utiliser`update-subscription-definition`.

AWS CLI

Pour mettre à jour le nom d'une définition d'abonnement

L'`update-subscription-definition`exemple suivant met à jour le nom de la définition d'abonnement spécifiée. Si vous souhaitez modifier les détails de l'abonnement, utilisez la `create-subscription-definition-version` commande pour créer une nouvelle version.

```
aws greengrass update-subscription-definition \
    --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \
    --name "ObsoleteSubscription"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir le titre dans le guide.

- Pour plus de détails sur l'API, reportez-vous UpdateSubscriptionDefinitionà la section Référence des AWS CLI commandes.

`update-thing-runtime-configuration`

L'exemple de code suivant montre comment utiliser`update-thing-runtime-configuration`.

AWS CLI

Pour activer la télémétrie dans la configuration d'exécution d'un noyau Greengrass

L'`update-thing-runtime-configuration`exemple suivant met à jour la configuration d'exécution d'un noyau Greengrass pour activer la télémétrie.

```
aws greengrass update-thing-runtime-configuration \
    --thing-name SampleGreengrassCore \
    --telemetry-configuration {\"Telemetry\":\"On\"}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration des paramètres de télémétrie dans le manuel IoT AWS Greengrass Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateThingRuntimeConfigurationà la section Référence des AWS CLI commandes.

AWS IoT Greengrass V2 exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Greengrass V2.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-service-role-to-account`

L'exemple de code suivant montre comment utiliser`associate-service-role-to-account`.

AWS CLI

Pour associer le rôle de service Greengrass à votre compte AWS

L'`associate-service-role-to-account`exemple suivant associe un rôle de service à AWS IoT Greengrass pour votre AWS compte.

```
aws greengrassv2 associate-service-role-to-account \
    --role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

Sortie :

```
{
    "associatedAt": "2022-01-19T19:21:53Z"
```

```
}
```

Pour plus d'informations, voir le rôle de service Greengrass dans le guide du développeur IoT AWS Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous AssociateServiceRoleToAccountà la section Référence des AWS CLI commandes.

`batch-associate-client-device-with-core-device`

L'exemple de code suivant montre comment utiliser`batch-associate-client-device-with-core-device`.

AWS CLI

Pour associer des appareils clients à un périphérique principal

L'`batch-associate-client-device-with-core-device`exemple suivant associe deux appareils clients à un périphérique principal.

```
aws greengrassv2 batch-associate-client-device-with-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Sortie :

```
{
    "errorEntries": []
}
```

Pour plus d'informations, consultez la section Interagir avec des appareils IoT locaux dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous BatchAssociateClientDeviceWithCoreDeviceà la section Référence des AWS CLI commandes.

`batch-disassociate-client-device-from-core-device`

L'exemple de code suivant montre comment utiliser`batch-disassociate-client-device-from-core-device`.

AWS CLI

Pour dissocier les appareils clients d'un périphérique principal

L'`batch-disassociate-client-device-from-core-device`exemple suivant dissocie deux appareils clients d'un périphérique principal.

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Sortie :

```
{
    "errorEntries": []
}
```

Pour plus d'informations, consultez la section Interagir avec des appareils IoT locaux dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous BatchDisassociateClientDeviceFromCoreDeviceà la section Référence des AWS CLI commandes.

`cancel-deployment`

L'exemple de code suivant montre comment utiliser`cancel-deployment`.

AWS CLI

Pour annuler un déploiement

L'`cancel-deployment`exemple suivant arrête un déploiement continu vers un groupe d'objets.

```
aws greengrassv2 cancel-deployment \
    --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "message": "SUCCESS"
}
```

Pour plus d'informations, consultez la section Annuler des déploiements dans le guide du AWS développeur IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous CancelDeploymentà la section Référence des AWS CLI commandes.

`create-component-version`

L'exemple de code suivant montre comment utiliser`create-component-version`.

AWS CLI

Exemple 1 : pour créer une version de composant à partir d'une recette

L'`create-component-version`exemple suivant crée une version d'un composant Hello World à partir d'un fichier de recette.

```
aws greengrassv2 create-component-version \
    --inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

Contenu de `com.example.HelloWorld-1.0.0.json` :

```
{
    "RecipeFormatVersion": "2020-01-25",
    "ComponentName": "com.example.HelloWorld",
    "ComponentVersion": "1.0.0",
    "ComponentDescription": "My first AWS IoT Greengrass component.",
    "ComponentPublisher": "Amazon",
    "ComponentConfiguration": {
        "DefaultConfiguration": {
            "Message": "world"
        }
    },
    "Manifests": [
        {
            "Platform": {
                "os": "linux"
            },
            "Lifecycle": {
                "Run": "echo 'Hello {configuration:/Message}'"
            }
        }
```

```
    ]
}
```

Sortie :

```
{
    "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "creationTimestamp": "2021-01-07T16:24:33.650000-08:00",
    "status": {
        "componentState": "REQUESTED",
        "message": "NONE",
        "errors": {}
    }
}
```

Pour plus d'informations, consultez les sections Création de composants personnalisés et Chargement de composants à déployer dans le Guide du développeur AWS IoT Greengrass V2.

Exemple 2 : pour créer une version de composant à partir d'une fonction AWS Lambda

L'`create-component-version`exemple suivant crée une version d'un composant Hello World à partir d'une fonction AWS Lambda.

```
aws greengrassv2 create-component-version \
    --cli-input-json file://lambda-function-component.json
```

Contenu de `lambda-function-component.json` :

```
{
    "lambdaFunction": {
        "lambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:HelloWorldPythonLambda:1",
        "componentName": "com.example.HelloWorld",
        "componentVersion": "1.0.0",
        "componentLambdaParameters": {
            "eventSources": [
                {
                    "topic": "hello/world/+",
                    "type": "IOT_CORE"
```

```
                }
            ]
        }
    }
}
```

Sortie :

```
{
    "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "creationTimestamp": "2021-01-07T17:05:27.347000-08:00",
    "status": {
        "componentState": "REQUESTED",
        "message": "NONE",
        "errors": {}
    }
}
```

Pour plus d'informations, consultez la section Exécuter les fonctions AWS Lambda dans le guide du développeur IoT AWS Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous CreateComponentVersionà la section Référence des AWS CLI commandes.

`create-deployment`

L'exemple de code suivant montre comment utiliser`create-deployment`.

AWS CLI

Exemple 1 : pour créer un déploiement

L'`create-deployment`exemple suivant déploie l'interface de ligne de commande AWS IoT Greengrass sur un appareil principal.

```
aws greengrassv2 create-deployment \
    --cli-input-json file://cli-deployment.json
```

Contenu de `cli-deployment.json` :

```
{
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/MyGreengrassCore",
    "deploymentName": "Deployment for MyGreengrassCore",
    "components": {
        "aws.greengrass.Cli": {
            "componentVersion": "2.0.3"
        }
    },
    "deploymentPolicies": {
        "failureHandlingPolicy": "DO_NOTHING",
        "componentUpdatePolicy": {
            "timeoutInSeconds": 60,
            "action": "NOTIFY_COMPONENTS"
        },
        "configurationValidationPolicy": {
            "timeoutInSeconds": 60
        }
    },
    "iotJobConfiguration": {}
}
```

Sortie :

```
{
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Création de déploiements dans le guide du AWS développeur IoT Greengrass V2.

Exemple 2 : pour créer un déploiement qui met à jour les configurations des composants

L'`create-deployment`exemple suivant déploie le composant du noyau AWS IoT Greengrass sur un groupe d'appareils principaux. Ce déploiement applique les mises à jour de configuration suivantes pour le composant Nucleus :

Réinitialisez les paramètres proxy des machines cibles à leurs paramètres par défaut sans proxy. Réinitialisez les paramètres MQTT des machines cibles à leurs valeurs par défaut.Définit les options JVM pour la JVM du noyau.Définit le niveau de journalisation du noyau.

```
aws greengrassv2 create-deployment \
```

```
    --cli-input-json file://nucleus-deployment.json
```

Contenu de `nucleus-deployment.json` :

```
{
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
    "deploymentName": "Deployment for MyGreengrassCoreGroup",
    "components": {
        "aws.greengrass.Nucleus": {
            "componentVersion": "2.0.3",
            "configurationUpdate": {
                "reset": [
                    "/networkProxy",
                    "/mqtt"
                ],
                "merge": "{\"jvmOptions\":\"-Xmx64m\",\"logging\":{\"level\":\"WARN
\"}}"
            }
        }
    },
    "deploymentPolicies": {
        "failureHandlingPolicy": "ROLLBACK",
        "componentUpdatePolicy": {
            "timeoutInSeconds": 60,
            "action": "NOTIFY_COMPONENTS"
        },
        "configurationValidationPolicy": {
            "timeoutInSeconds": 60
        }
    },
    "iotJobConfiguration": {}
}
```

Sortie :

```
{
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
}
```

Pour plus d'informations, consultez la section Création de déploiements et mise à jour des configurations de composants dans le guide du AWS développeur IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentà la section Référence des AWS CLI commandes.

`delete-component`

L'exemple de code suivant montre comment utiliser`delete-component`.

AWS CLI

Pour supprimer une version de composant

L'`delete-component`exemple suivant supprime un composant Hello World.

```
aws greengrassv2 delete-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous DeleteComponentà la section Référence des AWS CLI commandes.

`delete-core-device`

L'exemple de code suivant montre comment utiliser`delete-core-device`.

AWS CLI

Pour supprimer un appareil principal

L'`delete-core-device`exemple suivant supprime un appareil AWS IoT Greengrass central.

```
aws greengrassv2 delete-core-device \
    --core-device-thing-name MyGreengrassCore
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désinstaller le logiciel AWS IoT Greengrass Core dans le guide du développeur IoT AWS Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous DeleteCoreDeviceà la section Référence des AWS CLI commandes.

`describe-component`

L'exemple de code suivant montre comment utiliser`describe-component`.

AWS CLI

Pour décrire la version d'un composant

L'`describe-component`exemple suivant décrit un composant Hello World.

```
aws greengrassv2 describe-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

Sortie :

```
{
    "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "creationTimestamp": "2021-01-07T17:12:11.133000-08:00",
    "publisher": "Amazon",
    "description": "My first AWS IoT Greengrass component.",
    "status": {
        "componentState": "DEPLOYABLE",
        "message": "NONE",
        "errors": {}
    },
    "platforms": [
        {
            "attributes": {
                "os": "linux"
            }
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous DescribeComponentà la section Référence des AWS CLI commandes.

`disassociate-service-role-from-account`

L'exemple de code suivant montre comment utiliser`disassociate-service-role-from-account`.

AWS CLI

Pour dissocier le rôle de service Greengrass de votre compte AWS

L'`disassociate-service-role-from-account`exemple suivant dissocie le rôle de service Greengrass de AWS IoT Greengrass pour votre compte. AWS

```
aws greengrassv2 disassociate-service-role-from-account
```

Sortie :

```
{
    "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

Pour plus d'informations, voir le rôle de service Greengrass dans le guide du développeur IoT AWS Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous DisassociateServiceRoleFromAccountà la section Référence des AWS CLI commandes.

`get-component-version-artifact`

L'exemple de code suivant montre comment utiliser`get-component-version-artifact`.

AWS CLI

Pour obtenir une URL permettant de télécharger un artefact de composant

L'`get-component-version-artifact`exemple suivant obtient une URL pour télécharger le fichier JAR du composant de console de débogage local.

```
aws greengrassv2 get-component-version-artifact \
    --arn arn:aws:greengrass:us-
west-2:aws:components:aws.greengrass.LocalDebugConsole:versions:2.0.3 \
    --artifact-name "Uvt6ZEzQ9TKiAuLbfXBX_APdY0TWks3uc46tHFHTzBM=/
aws.greengrass.LocalDebugConsole.jar"
```

Sortie :

```
{
    "preSignedUrl": "https://evergreencomponentmanageme-
artifactbucket7410c9ef-g18n1iya8kwr.s3.us-west-2.amazonaws.com/public/
aws.greengrass.LocalDebugConsole/2.0.3/s3/ggv2-component-releases-prod-pdx/
EvergreenHttpDebugView/2ffc496ba41b39568968b22c582b4714a937193ee7687a45527238e696672521/
aws.greengrass.LocalDebugConsole/aws.greengrass.LocalDebugConsole.jar?X-Amz-
Security-Token=KwfLKSdEXAMPLE..."
}
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous GetComponentVersionArtifactà la section Référence des AWS CLI commandes.

`get-component`

L'exemple de code suivant montre comment utiliser`get-component`.

AWS CLI

Exemple 1 : pour télécharger la recette d'un composant au format YAML (Linux, macOS ou Unix)

L'`get-component`exemple suivant télécharge la recette d'un composant Hello World dans un fichier au format YAML. Cette commande exécute les opérations suivantes :

Utilise les `--query` paramètres `--output` et pour contrôler le résultat de la commande. Ces paramètres extraient le blob de recettes de la sortie de la commande. Pour plus d'informations sur le contrôle de la sortie, voir Contrôle de la sortie des commandes dans le Guide de l'utilisateur de l'interface de ligne de AWS commande. Utilise l'`base64`utilitaire. Cet utilitaire décode le

blob extrait selon le texte d'origine. Le blob renvoyé par une `get-component` commande réussie est du texte codé en base64. Vous devez décoder ce blob pour obtenir le texte original. Enregistre le texte décodé dans un fichier. La dernière section de la commande (> `com.example.HelloWorld-1.0.0.json`) enregistre le texte décodé dans un fichier.

```
aws greengrassv2 get-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0 \
    --recipe-output-format YAML \
    --query recipe \
    --output text | base64 --decode > com.example.HelloWorld-1.0.0.json
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

Exemple 2 : pour télécharger la recette d'un composant au format YAML (Windows CMD)

L'`get-component`exemple suivant télécharge la recette d'un composant Hello World dans un fichier au format YAML. Cette commande utilise l'`certutil`utilitaire.

```
aws greengrassv2 get-component ^
    --arn arn:aws:greengrass:us-
west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 ^
    --recipe-output-format YAML ^
    --query recipe ^
    --output text > com.example.HelloWorld-1.0.0.yaml.b64

certutil -
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

Exemple 3 : pour télécharger la recette d'un composant au format YAML (Windows PowerShell)

L'`get-component`exemple suivant télécharge la recette d'un composant Hello World dans un fichier au format YAML. Cette commande utilise l'`certutil`utilitaire.

```
aws greengrassv2 get-component `
    --arn arn:aws:greengrass:us-
west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 `
    --recipe-output-format YAML `
```

```
    --query recipe `
    --output text > com.example.HelloWorld-1.0.0.yaml.b64

certutil -
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous GetComponentà la section Référence des AWS CLI commandes.

`get-connectivity-info`

L'exemple de code suivant montre comment utiliser`get-connectivity-info`.

AWS CLI

Pour obtenir les informations de connectivité d'un appareil Greengrass Core

L'`get-connectivity-info`exemple suivant permet d'obtenir les informations de connectivité d'un appareil Greengrass Core. Les appareils clients utilisent ces informations pour se connecter au broker MQTT qui s'exécute sur ce périphérique principal.

```
aws greengrassv2 get-connectivity-info \
    --thing-name MyGreengrassCore
```

Sortie :

```
{
    "connectivityInfo": [
        {
            "id": "localIP_192.0.2.0",
            "hostAddress": "192.0.2.0",
            "portNumber": 8883
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les principaux points de terminaison des appareils dans le guide du AWS développeur IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous GetConnectivityInfoà la section Référence des AWS CLI commandes.

`get-core-device`

L'exemple de code suivant montre comment utiliser`get-core-device`.

AWS CLI

Pour obtenir un appareil principal

L'`get-core-device`exemple suivant permet d'obtenir des informations sur un AWS appareil IoT Greengrass central.

```
aws greengrassv2 get-core-device \
    --core-device-thing-name MyGreengrassCore
```

Sortie :

```
{
    "coreDeviceThingName": "MyGreengrassCore",
    "coreVersion": "2.0.3",
    "platform": "linux",
    "architecture": "amd64",
    "status": "HEALTHY",
    "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Vérifier l'état de l'appareil principal dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous GetCoreDeviceà la section Référence des AWS CLI commandes.

`get-deployment`

L'exemple de code suivant montre comment utiliser`get-deployment`.

AWS CLI

Pour obtenir un déploiement

L'`get-deployment`exemple suivant fournit des informations sur le déploiement du composant AWS IoT Greengrass nucleus sur un groupe d'appareils principaux.

```
aws greengrassv2 get-deployment \
    --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
    "revisionId": "14",
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "deploymentName": "Deployment for MyGreengrassCoreGroup",
    "deploymentStatus": "ACTIVE",
    "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222",
    "components": {
        "aws.greengrass.Nucleus": {
            "componentVersion": "2.0.3",
            "configurationUpdate": {
                "merge": "{\"jvmOptions\":\"-Xmx64m\",\"logging\":{\"level\":\"WARN
\"}}",
                "reset": [
                    "/networkProxy",
                    "/mqtt"
                ]
            }
        }
    },
    "deploymentPolicies": {
        "failureHandlingPolicy": "ROLLBACK",
        "componentUpdatePolicy": {
            "timeoutInSeconds": 60,
            "action": "NOTIFY_COMPONENTS"
        },
        "configurationValidationPolicy": {
            "timeoutInSeconds": 60
        }
    },
    "iotJobConfiguration": {},
    "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
```

```
    "isLatestForTarget": false,
    "tags": {}
}
```

Pour plus d'informations, consultez la section Déployer des composants sur des appareils dans le AWS guide du développeur IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous GetDeploymentà la section Référence des AWS CLI commandes.

`get-service-role-for-account`

L'exemple de code suivant montre comment utiliser`get-service-role-for-account`.

AWS CLI

Pour obtenir le rôle de service Greengrass associé à votre compte AWS

L'`get-service-role-for-account`exemple suivant montre le rôle de service associé à AWS IoT Greengrass pour votre AWS compte.

```
aws greengrassv2 get-service-role-for-account
```

Sortie :

```
{
    "associatedAt": "2022-01-19T19:21:53Z",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Pour plus d'informations, voir le rôle de service Greengrass dans le guide du développeur IoT AWS Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous GetServiceRoleForAccountà la section Référence des AWS CLI commandes.

`list-client-devices-associated-with-core-device`

L'exemple de code suivant montre comment utiliser`list-client-devices-associated-with-core-device`.

AWS CLI

Pour répertorier les appareils clients associés à un périphérique principal

L'`list-client-devices-associated-with-core-device`exemple suivant répertorie tous les appareils clients associés à un périphérique principal.

```
aws greengrassv2 list-client-devices-associated-with-core-device \
  --core-device-thing-name MyTestGreengrassCore
```

Sortie :

```
{
    "associatedClientDevices": [
        {
            "thingName": "MyClientDevice2",
            "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
        },
        {
            "thingName": "MyClientDevice1",
            "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Interagir avec des appareils IoT locaux dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListClientDevicesAssociatedWithCoreDeviceà la section Référence des AWS CLI commandes.

`list-component-versions`

L'exemple de code suivant montre comment utiliser`list-component-versions`.

AWS CLI

Pour répertorier les versions d'un composant

L'`list-component-versions`exemple suivant répertorie toutes les versions d'un composant Hello World.

```
aws greengrassv2 list-component-versions \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld
```

Sortie :

```
{
    "componentVersions": [
        {
            "componentName": "com.example.HelloWorld",
            "componentVersion": "1.0.1",
            "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1"
        },
        {
            "componentName": "com.example.HelloWorld",
            "componentVersion": "1.0.0",
            "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListComponentVersionsà la section Référence des AWS CLI commandes.

`list-components`

L'exemple de code suivant montre comment utiliser`list-components`.

AWS CLI

Pour répertorier les composants

L'`list-components`exemple suivant répertorie chaque composant et sa dernière version définis dans votre AWS compte dans la région actuelle.

```
aws greengrassv2 list-components
```

Sortie :

```
{
    "components": [
        {
            "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld",
            "componentName": "com.example.HelloWorld",
            "latestVersion": {
                "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1",
                "componentVersion": "1.0.1",
                "creationTimestamp": "2021-01-08T16:51:07.352000-08:00",
                "description": "My first AWS IoT Greengrass component.",
                "publisher": "Amazon",
                "platforms": [
                    {
                        "attributes": {
                            "os": "linux"
                        }
                    }
                ]
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les composants dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListComponentsà la section Référence des AWS CLI commandes.

`list-core-devices`

L'exemple de code suivant montre comment utiliser`list-core-devices`.

AWS CLI

Pour répertorier les principaux appareils

L'`list-core-devices`exemple suivant répertorie les principaux appareils AWS IoT Greengrass de votre AWS compte dans la région actuelle.

```
aws greengrassv2 list-core-devices
```

Sortie :

```
{
    "coreDevices": [
        {
            "coreDeviceThingName": "MyGreengrassCore",
            "status": "HEALTHY",
            "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Vérifier l'état de l'appareil principal dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListCoreDevicesà la section Référence des AWS CLI commandes.

`list-deployments`

L'exemple de code suivant montre comment utiliser`list-deployments`.

AWS CLI

Pour répertorier les déploiements

L'`list-deployments`exemple suivant répertorie la dernière révision de chaque déploiement défini dans votre AWS compte dans la région actuelle.

```
aws greengrassv2 list-deployments
```

Sortie :

```
{
    "deployments": [
        {
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
            "revisionId": "14",
```

```
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "deploymentName": "Deployment for MyGreengrassCoreGroup",
            "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
            "deploymentStatus": "ACTIVE",
            "isLatestForTarget": false
        },
        {
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
            "revisionId": "1",
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "deploymentName": "Deployment for MyGreengrassCore",
            "creationTimestamp": "2021-01-06T16:10:42.407000-08:00",
            "deploymentStatus": "COMPLETED",
            "isLatestForTarget": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Déployer des composants sur des appareils dans le AWS guide du développeur IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListDeploymentsà la section Référence des AWS CLI commandes.

`list-effective-deployments`

L'exemple de code suivant montre comment utiliser`list-effective-deployments`.

AWS CLI

Pour répertorier les tâches de déploiement

L'`list-effective-deployments`exemple suivant répertorie les déploiements qui s'appliquent à un appareil principal AWS IoT Greengrass.

```
aws greengrassv2 list-effective-deployments \
    --core-device-thing-name MyGreengrassCore
```

Sortie :

```
{
    "effectiveDeployments": [
```

```
        {
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "deploymentName": "Deployment for MyGreengrassCore",
            "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
            "coreDeviceExecutionStatus": "COMPLETED",
            "reason": "SUCCESSFUL",
            "creationTimestamp": "2021-01-06T16:10:42.442000-08:00",
            "modifiedTimestamp": "2021-01-08T17:21:27.830000-08:00"
        },
        {
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "deploymentName": "Deployment for MyGreengrassCoreGroup",
            "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-
cdef-EXAMPLE44444",
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
            "coreDeviceExecutionStatus": "SUCCEEDED",
            "reason": "SUCCESSFUL",
            "creationTimestamp": "2021-01-07T17:19:20.394000-08:00",
            "modifiedTimestamp": "2021-01-07T17:21:20.721000-08:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Vérifier l'état de l'appareil principal dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListEffectiveDeploymentsà la section Référence des AWS CLI commandes.

`list-installed-components`

L'exemple de code suivant montre comment utiliser`list-installed-components`.

AWS CLI

Pour répertorier les composants installés sur un périphérique principal

L'`list-installed-components`exemple suivant répertorie les composants installés sur un appareil AWS IoT Greengrass central.

```
aws greengrassv2 list-installed-components \
    --core-device-thing-name MyGreengrassCore
```

Sortie :

```
{
    "installedComponents": [
        {
            "componentName": "aws.greengrass.Cli",
            "componentVersion": "2.0.3",
            "lifecycleState": "RUNNING",
            "isRoot": true
        },
        {
            "componentName": "aws.greengrass.Nucleus",
            "componentVersion": "2.0.3",
            "lifecycleState": "FINISHED",
            "isRoot": true
        }
    ]
}
```

Pour plus d'informations, consultez la section Vérifier l'état de l'appareil principal dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListInstalledComponentsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'un appareil AWS IoT Greengrass central.

```
aws greengrassv2 list-tags-for-resource \
```

```
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore
```

Sortie :

```
{
    "tags": {
        "Owner": "richard-roe"
    }
}
```

Pour plus d'informations, consultez la section Marquer vos ressources dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Ajout d'une balise à une ressource

L'`tag-resource`exemple suivant ajoute une étiquette de propriétaire à un appareil AWS IoT Greengrass central. Vous pouvez utiliser cette balise pour contrôler l'accès à l'appareil principal en fonction de son propriétaire.

```
aws greengrassv2 tag-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore \
    --tags Owner=richard-roe
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquer vos ressources dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime une étiquette de propriétaire d'un appareil AWS IoT Greengrass central.

```
aws iotsitewise untag-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore \
    --tag-keys Owner
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquer vos ressources dans le guide du développeur AWS IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-connectivity-info`

L'exemple de code suivant montre comment utiliser`update-connectivity-info`.

AWS CLI

Pour mettre à jour les informations de connectivité d'un appareil Greengrass Core

L'`update-connectivity-info`exemple suivant permet d'obtenir les informations de connectivité d'un appareil Greengrass Core. Les appareils clients utilisent ces informations pour se connecter au broker MQTT qui s'exécute sur ce périphérique principal.

```
aws greengrassv2 update-connectivity-info \
    --thing-name MyGreengrassCore \
    --cli-input-json file://core-device-connectivity-info.json
```

Contenu de `core-device-connectivity-info.json` :

```
{
    "connectivityInfo": [
        {
            "hostAddress": "192.0.2.0",
            "portNumber": 8883,
            "id": "localIP_192.0.2.0"
        }
    ]
}
```

Sortie :

```
{
    "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Gérer les principaux points de terminaison des appareils dans le guide du AWS développeur IoT Greengrass V2.

- Pour plus de détails sur l'API, reportez-vous UpdateConnectivityInfoà la section Référence des AWS CLI commandes.

AWS IoT Jobs SDK release exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Jobs SDK release.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-job-execution`

L'exemple de code suivant montre comment utiliser`describe-job-execution`.

AWS CLI

Pour obtenir les détails de l'exécution d'une tâche

L'`describe-job-execution`exemple suivant récupère les détails de la dernière exécution de la tâche et de l'objet spécifiés.

```
aws iot-jobs-data describe-job-execution \
    --job-id SampleJob \
    --thing-name MotionSensor1 \
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Sortie :

```
{
    "execution": {
        "approximateSecondsBeforeTimedOut": 88,
        "executionNumber": 2939653338,
        "jobId": "SampleJob",
        "lastUpdatedAt": 1567701875.743,
        "queuedAt": 1567701902.444,
        "status": "QUEUED",
        "thingName": "MotionSensor1 ",
        "versionNumber": 3
   }
}
```

Pour plus d'informations, consultez la section Appareils et emplois du Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeJobExecutionà la section Référence des AWS CLI commandes.

`get-pending-job-executions`

L'exemple de code suivant montre comment utiliser`get-pending-job-executions`.

AWS CLI

Pour obtenir la liste de toutes les tâches qui n'ont pas le statut de terminal pour un objet

L'`get-pending-job-executions`exemple suivant affiche une liste de toutes les tâches qui ne sont pas dans un état terminal pour l'objet spécifié.

```
aws iot-jobs-data get-pending-job-executions \
    --thing-name MotionSensor1
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Sortie :

```
{
    "inProgressJobs": [
    ],
    "queuedJobs": [
        {
            "executionNumber": 2939653338,
            "jobId": "SampleJob",
            "lastUpdatedAt": 1567701875.743,
            "queuedAt": 1567701902.444,
            "versionNumber": 3
      }
    ]
}
```

Pour plus d'informations, consultez la section Appareils et emplois du Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetPendingJobExecutionsà la section Référence des AWS CLI commandes.

`start-next-pending-job-execution`

L'exemple de code suivant montre comment utiliser`start-next-pending-job-execution`.

AWS CLI

Pour obtenir et démarrer la prochaine exécution d'une tâche en attente pour un objet

L'`start-next-pending-job-execution`exemple suivant récupère et démarre l'exécution de la tâche suivante dont le statut est IN_PROGRESS ou QUEUED pour l'objet spécifié.

```
aws iot-jobs-data start-next-pending-job-execution \
    --thing-name MotionSensor1
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Sortie :

```
{
    "execution": {
        "approximateSecondsBeforeTimedOut": 88,
        "executionNumber": 2939653338,
        "jobId": "SampleJob",
        "lastUpdatedAt": 1567714853.743,
        "queuedAt": 1567701902.444,
        "startedAt": 1567714871.690,
        "status": "IN_PROGRESS",
        "thingName": "MotionSensor1 ",
        "versionNumber": 3
   }
}
```

Pour plus d'informations, consultez la section Appareils et emplois du Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous StartNextPendingJobExecutionà la section Référence des AWS CLI commandes.

`update-job-execution`

L'exemple de code suivant montre comment utiliser`update-job-execution`.

AWS CLI

Pour mettre à jour le statut de l'exécution d'une tâche

L'`update-job-execution`exemple suivant met à jour le statut de la tâche et de l'objet spécifiés.

```
aws iot-jobs-data update-job-execution \
    --job-id SampleJob \
```

```
    --thing-name MotionSensor1 \
    --status REMOVED \
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Sortie :

```
{
    "executionState": {
        "status": "REMOVED",
        "versionNumber": 3
    },
}
```

Pour plus d'informations, consultez la section Appareils et emplois du Guide du développeur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateJobExecutionà la section Référence des AWS CLI commandes.

AWS IoT SiteWise exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT SiteWise.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-assets`

L'exemple de code suivant montre comment utiliser`associate-assets`.

AWS CLI

Pour associer un actif enfant à un actif parent

L'`associate-assets`exemple suivant associe un actif d'éolienne à un actif de parc éolien, où le modèle d'actif d'éolienne existe sous forme de hiérarchie dans le modèle d'actif de parc éolien.

```
aws iotsitewise associate-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \
    --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Associer des actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous AssociateAssetsà la section Référence des AWS CLI commandes.

`batch-associate-project-assets`

L'exemple de code suivant montre comment utiliser`batch-associate-project-assets`.

AWS CLI

Pour associer un actif à un projet

L'`batch-associate-project-assets`exemple suivant associe un actif de parc éolien à un projet.

```
aws iotsitewise batch-associate-project-assets \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des actifs aux projets dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous BatchAssociateProjectAssetsà la section Référence des AWS CLI commandes.

batch-disassociate-project-assets

L'exemple de code suivant montre comment utiliserbatch-disassociate-project-assets.

AWS CLI

Pour dissocier un actif d'un projet

L'batch-disassociate-project-assetsexemple suivant dissocie un actif d'un parc éolien d'un projet.

```
aws iotsitewise batch-disassociate-project-assets \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des actifs aux projets dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous BatchDisassociateProjectAssetsà la section Référence des AWS CLI commandes.

batch-put-asset-property-value

L'exemple de code suivant montre comment utiliserbatch-put-asset-property-value.

AWS CLI

Pour envoyer des données aux propriétés des actifs

L'batch-put-asset-property-valueexemple suivant envoie des données de puissance et de température aux propriétés des actifs identifiées par des alias de propriété.

```
aws iotsitewise batch-put-asset-property-value \
    --cli-input-json file://batch-put-asset-property-value.json
```

Contenu de batch-put-asset-property-value.json :

```
{
    "entries": [
```

```
        {
            "entryId": "1575691200-company-windfarm-3-turbine-7-power",
            "propertyAlias": "company-windfarm-3-turbine-7-power",
            "propertyValues": [
                {
                    "value": {
                        "doubleValue": 4.92
                    },
                    "timestamp": {
                        "timeInSeconds": 1575691200
                    },
                    "quality": "GOOD"
                }
            ]
        },
        {
            "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
            "propertyAlias": "company-windfarm-3-turbine-7-temperature",
            "propertyValues": [
                {
                    "value": {
                        "integerValue": 38
                    },
                    "timestamp": {
                        "timeInSeconds": 1575691200
                    }
                }
            ]
        }
    ]
}
```

Sortie :

```
{
    "errorEntries": []
}
```

Pour plus d'informations, consultez la section Ingestion de données à l'aide de l' SiteWise API AWS IoT dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous BatchPutAssetPropertyValueà la section Référence des AWS CLI commandes.

`create-access-policy`

L'exemple de code suivant montre comment utiliser`create-access-policy`.

AWS CLI

Exemple 1 : accorder à un utilisateur un accès administratif à un portail

L'`create-access-policy`exemple suivant crée une politique d'accès qui accorde à un utilisateur un accès administratif à un portail Web pour une société de parc éolien.

```
aws iotsitewise create-access-policy \
    --cli-input-json file://create-portal-administrator-access-policy.json
```

Contenu de `create-portal-administrator-access-policy.json` :

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
        "portal": {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
        }
    }
}
```

Sortie :

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE"
}
```

Pour plus d'informations, consultez la section Ajouter ou supprimer des administrateurs de portail dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

Exemple 2 : Pour accorder à un utilisateur un accès en lecture seule à un projet

L'`create-access-policy`exemple suivant crée une politique d'accès qui accorde à un utilisateur un accès en lecture seule à un projet de parc éolien.

```
aws iotsitewise create-access-policy \
    --cli-input-json file://create-project-viewer-access-policy.json
```

Contenu de `create-project-viewer-access-policy.json` :

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "VIEWER",
    "accessPolicyResource": {
        "project": {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
        }
    }
}
```

Sortie :

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"
}
```

Pour plus d'informations, consultez la section Affectation de visualiseurs de projet dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous CreateAccessPolicyà la section Référence des AWS CLI commandes.

`create-asset-model`

L'exemple de code suivant montre comment utiliser`create-asset-model`.

AWS CLI

Pour créer un modèle d'actifs

L'`create-asset-model`exemple suivant crée un modèle d'actif qui définit une éolienne avec les propriétés suivantes :

Numéro de série - Le numéro de série d'une éolienne L'énergie produite par une éolienne - Le flux de données de puissance généré par une éolienne Température C - Le flux de données de température d'une éolienne en CelsiusTemperature F - Les points de données de température cartographiés de Celsius à Fahrenheit

```
aws iotsitewise create-asset-model \
    --cli-input-json file://create-wind-turbine-model.json
```

Contenu de `create-wind-turbine-model.json` :

```
{
    "assetModelName": "Wind Turbine Model",
    "assetModelDescription": "Represents a wind turbine",
    "assetModelProperties": [
        {
            "name": "Serial Number",
            "dataType": "STRING",
            "type": {
                "attribute": {}
            }
        },
        {
            "name": "Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "measurement": {}
            }
        },
        {
            "name": "Temperature C",
            "dataType": "DOUBLE",
            "unit": "Celsius",
            "type": {
                "measurement": {}
            }
```

```
        },
        {
            "name": "Temperature F",
            "dataType": "DOUBLE",
            "unit": "Fahrenheit",
            "type": {
                "transform": {
                    "expression": "temp_c * 9 / 5 + 32",
                    "variables": [
                        {
                            "name": "temp_c",
                            "value": {
                                "propertyId": "Temperature C"
                            }
                        }
                    ]
                }
            }
        },
        {
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "metric": {
                    "expression": "sum(power)",
                    "variables": [
                        {
                            "name": "power",
                            "value": {
                                "propertyId": "Generated Power"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
                        }
                    }
                }
            }
        }
    ]
```

```
}
```

Sortie :

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}
```

Pour plus d'informations, consultez la section Définition des modèles d'actifs dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateAssetModelà la section Référence des AWS CLI commandes.

`create-asset`

L'exemple de code suivant montre comment utiliser`create-asset`.

AWS CLI

Pour créer un actif

L'`create-asset`exemple suivant crée un actif d'éolienne à partir d'un modèle d'actif d'éolienne.

```
aws iotsitewise create-asset \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --asset-name "Wind Turbine 1"
```

Sortie :

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetStatus": {
        "state": "CREATING"
    }
```

```
}
```

Pour plus d'informations, consultez la section Création d'actifs dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateAsseta la section Référence des AWS CLI commandes.

`create-dashboard`

L'exemple de code suivant montre comment utiliser`create-dashboard`.

AWS CLI

Pour créer un tableau de bord

L'`create-dashboard`exemple suivant crée un tableau de bord avec un graphique linéaire qui affiche la puissance totale produite pour un parc éolien.

```
aws iotsitewise create-dashboard \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --dashboard-name "Wind Farm" \
    --dashboard-definition file://create-wind-farm-dashboard.json
```

Contenu de `create-wind-farm-dashboard.json` :

```
{
    "widgets": [
        {
            "type": "monitor-line-chart",
            "title": "Generated Power",
            "x": 0,
            "y": 0,
            "height": 3,
            "width": 3,
            "metrics": [
                {
                    "label": "Power",
                    "type": "iotsitewise",
                    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
                    "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"
                }
```

```
            ]
        }
    ]
}
```

Sortie :

```
{
    "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE"
}
```

Pour plus d'informations, consultez la section Création de tableaux de bord (CLI) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateDashboardà la section Référence des AWS CLI commandes.

`create-gateway`

L'exemple de code suivant montre comment utiliser`create-gateway`.

AWS CLI

Pour créer une passerelle

L'`create-gateway`exemple suivant crée une passerelle qui s'exécute sur AWS IoT Greengrass.

```
aws iotsitewise create-gateway \
    --gateway-name ExampleCorpGateway \
    --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}
```

Sortie :

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

Pour plus d'informations, consultez la section Configuration d'une passerelle dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateGatewayà la section Référence des AWS CLI commandes.

`create-portal`

L'exemple de code suivant montre comment utiliser`create-portal`.

AWS CLI

Pour créer un portail

L'`create-portal`exemple suivant crée un portail Web pour une société de parc éolien. Vous ne pouvez créer des portails que dans la région où vous avez activé l'authentification AWS unique.

```
aws iotsitewise create-portal \
    --portal-name WindFarmPortal \
    --portal-description "A portal that contains wind farm projects for Example
 Corp." \
    --portal-contact-email support@example.com \
    --role-arn arn:aws:iam::123456789012:role/service-role/
MySiteWiseMonitorServiceRole
```

Sortie :

```
{
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/
a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
    "portalStatus": {
        "state": "CREATING"
    },
    "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"
}
```

Pour plus d'informations, consultez Getting started with AWS IoT SiteWise Monitor dans le guide de l' SiteWise utilisateur de l'AWS IoT et Enabling AWS SSO dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreatePortalà la section Référence des AWS CLI commandes.

`create-project`

L'exemple de code suivant montre comment utiliser`create-project`.

AWS CLI

Pour créer un projet

L'`create-project`exemple suivant crée un projet de parc éolien.

```
aws iotsitewise create-project \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \
    --project-name "Wind Farm 1" \
    --project-description "Contains asset visualizations for Wind Farm #1 for
 Example Corp."
```

Sortie :

```
{
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
}
```

Pour plus d'informations, consultez la section Création de projets dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous CreateProjectà la section Référence des AWS CLI commandes.

`delete-access-policy`

L'exemple de code suivant montre comment utiliser`delete-access-policy`.

AWS CLI

Pour révoquer l'accès d'un utilisateur à un projet ou à un portail

L'`delete-access-policy`exemple suivant supprime une politique d'accès qui accorde à un utilisateur un accès administratif à un portail.

```
aws iotsitewise delete-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter ou supprimer des administrateurs de portail dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteAccessPolicyà la section Référence des AWS CLI commandes.

`delete-asset-model`

L'exemple de code suivant montre comment utiliser`delete-asset-model`.

AWS CLI

Pour supprimer un modèle de ressource

L'`delete-asset-model`exemple suivant supprime un modèle d'actif d'éolienne.

```
aws iotsitewise delete-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
    "assetModelStatus": {
        "state": "DELETING"
    }
}
```

Pour plus d'informations, consultez la section Suppression de modèles d'actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteAssetModelà la section Référence des AWS CLI commandes.

`delete-asset`

L'exemple de code suivant montre comment utiliser`delete-asset`.

AWS CLI

Pour supprimer un actif

L'`delete-asset`exemple suivant supprime un actif d'éolienne.

```
aws iotsitewise delete-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Sortie :

```
{
    "assetStatus": {
        "state": "DELETING"
    }
}
```

Pour plus d'informations, consultez la section Suppression d'actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteAssetà la section Référence des AWS CLI commandes.

`delete-dashboard`

L'exemple de code suivant montre comment utiliser`delete-dashboard`.

AWS CLI

Pour supprimer un tableau de bord

L'`delete-dashboard`exemple suivant supprime un tableau de bord d'éolienne.

```
aws iotsitewise delete-dashboard \
    --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression de tableaux de bord dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous DeleteDashboardà la section Référence des AWS CLI commandes.

`delete-gateway`

L'exemple de code suivant montre comment utiliser`delete-gateway`.

AWS CLI

Pour supprimer une passerelle

L'`delete-gateway`exemple suivant supprime une passerelle.

```
aws iotsitewise delete-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ingestion de données à l'aide d'une passerelle dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteGatewayà la section Référence des AWS CLI commandes.

`delete-portal`

L'exemple de code suivant montre comment utiliser`delete-portal`.

AWS CLI

Pour supprimer un portail

L'`delete-portal`exemple suivant supprime le portail Web d'une société de parcs éoliens.

```
aws iotsitewise delete-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Sortie :

```
{
    "portalStatus": {
        "state": "DELETING"
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un portail dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeletePortalà la section Référence des AWS CLI commandes.

`delete-project`

L'exemple de code suivant montre comment utiliser`delete-project`.

AWS CLI

Pour supprimer un projet

L'`delete-project`exemple suivant supprime un projet de parc éolien.

```
aws iotsitewise delete-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression de projets dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous DeleteProjectà la section Référence des AWS CLI commandes.

`describe-access-policy`

L'exemple de code suivant montre comment utiliser`describe-access-policy`.

AWS CLI

Pour décrire une politique d'accès

L'`describe-access-policy`exemple suivant décrit une politique d'accès qui accorde à un utilisateur un accès administratif à un portail Web pour une société de parc éolien.

```
aws iotsitewise describe-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

Sortie :

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyResource": {
        "portal": {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyCreationDate": "2020-02-20T22:35:15.552880124Z",
    "accessPolicyLastUpdateDate": "2020-02-20T22:35:15.552880124Z"
}
```

Pour plus d'informations, consultez la section Ajouter ou supprimer des administrateurs de portail dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeAccessPolicyà la section Référence des AWS CLI commandes.

`describe-asset-model`

L'exemple de code suivant montre comment utiliser`describe-asset-model`.

AWS CLI

Pour décrire un modèle d'actifs

L'`describe-asset-model`exemple suivant décrit un modèle d'actifs de parc éolien.

```
aws iotsitewise describe-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Sortie :

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelName": "Wind Farm Model",
    "assetModelDescription": "Represents a wind farm that comprises many wind
 turbines",
    "assetModelProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "metric": {
                    "expression": "sum(power)",
                    "variables": [
                        {
                            "name": "power",
                            "value": {
                                "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
                                "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
                        }
                    }
                }
            }
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
```

```
            "dataType": "STRING",
            "type": {
                "attribute": {
                    "defaultValue": " "
                }
            }
        }
    ],
    "assetModelHierarchies": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines",
            "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ],
    "assetModelCreationDate": 1575671284.0,
    "assetModelLastUpdateDate": 1575671988.0,
    "assetModelStatus": {
        "state": "ACTIVE"
    }
}
```

Pour plus d'informations, consultez la section Décrire un modèle d'actif spécifique dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeAssetModelà la section Référence des AWS CLI commandes.

`describe-asset-property`

L'exemple de code suivant montre comment utiliser`describe-asset-property`.

AWS CLI

Pour décrire la propriété d'un actif

L'`describe-asset-property`exemple suivant décrit la propriété énergétique totale d'un actif de parc éolien.

```
aws iotsitewise describe-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE
```

Sortie :

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetName": "Wind Farm 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetProperty": {
        "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
        "name": "Total Generated Power",
        "notification": {
            "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE/properties/
a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "state": "DISABLED"
        },
        "dataType": "DOUBLE",
        "unit": "kW",
        "type": {
            "metric": {
                "expression": "sum(power)",
                "variables": [
                    {
                        "name": "power",
                        "value": {
                            "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE",
                            "hierarchyId": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE"
                        }
                    }
                ],
                "window": {
                    "tumbling": {
                        "interval": "1h"
                    }
                }
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Décrire une propriété d'actif spécifique dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeAssetPropertyà la section Référence des AWS CLI commandes.

`describe-asset`

L'exemple de code suivant montre comment utiliser`describe-asset`.

AWS CLI

Pour décrire un actif

L'`describe-asset`exemple suivant décrit un actif de parc éolien.

```
aws iotsitewise describe-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Sortie :

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetName": "Wind Farm 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
            "dataType": "STRING"
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW"
        }
    ],
    "assetHierarchies": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines"
        }
```

```
    ],
    "assetCreationDate": 1575672453.0,
    "assetLastUpdateDate": 1575672453.0,
    "assetStatus": {
        "state": "ACTIVE"
    }
}
```

Pour plus d'informations, consultez la section Décrire un actif spécifique dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeAssetà la section Référence des AWS CLI commandes.

`describe-dashboard`

L'exemple de code suivant montre comment utiliser`describe-dashboard`.

AWS CLI

Pour décrire un tableau de bord

L'`describe-dashboard`exemple suivant décrit le tableau de bord du parc éolien spécifié.

```
aws iotsitewise describe-dashboard \
    --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

Sortie :

```
{
    "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardName": "Wind Farm",
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "dashboardDefinition": "{\"widgets\":[{\"type\":\"monitor-line-chart\",\"title
\":\"Generated Power\",\"x\":0,\"y\":0,\"height\":3,\"width\":3,\"metrics\":
[{\"label\":\"Power\",\"type\":\"iotsitewise\",\"assetId\":\"a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE\",\"propertyId\":\"a1b2c3d4-5678-90ab-cdef-99999EXAMPLE\"}]}]}",
    "dashboardCreationDate": "2020-05-01T20:32:12.228476348Z",
    "dashboardLastUpdateDate": "2020-05-01T20:32:12.228476348Z"
```

```
}
```

Pour plus d'informations, consultez la section Affichage des tableaux de bord dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous DescribeDashboardà la section Référence des AWS CLI commandes.

`describe-gateway-capability-configuration`

L'exemple de code suivant montre comment utiliser`describe-gateway-capability-configuration`.

AWS CLI

Pour décrire une fonctionnalité de passerelle

L'`describe-gateway-capability-configuration`exemple suivant décrit une fonctionnalité de source OPC-UA.

```
aws iotsitewise describe-gateway-capability-configuration \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --capability-namespace "iotsitewise:opcuacollector:1"
```

Sortie :

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "capabilityNamespace": "iotsitewise:opcuacollector:1",
    "capabilityConfiguration": "{\"sources\":[{\"name\":\"Wind Farm #1\",
\"endpoint\":{\"certificateTrust\":{\"type\":\"TrustAny\"},\"endpointUri
\":\"opc.tcp://203.0.113.0:49320\",\"securityPolicy\":\"BASIC256\",
\"messageSecurityMode\":\"SIGN_AND_ENCRYPT\",\"identityProvider\":
{\"type\":\"Username\",\"usernameSecretArn\":\"arn:aws:secretsmanager:us-
east-1:123456789012:secret:greengrass-factory1-auth-3QNDmM\"},\"nodeFilterRules\":
[]},\"measurementDataStreamPrefix\":\"\"}]}",
    "capabilitySyncStatus": "IN_SYNC"
}
```

Pour plus d'informations, consultez la section Configuration des sources de données dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeGatewayCapabilityConfigurationà la section Référence des AWS CLI commandes.

`describe-gateway`

L'exemple de code suivant montre comment utiliser`describe-gateway`.

AWS CLI

Pour décrire une passerelle

L'`describe-gateway`exemple suivant décrit une passerelle.

```
aws iotsitewise describe-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Sortie :

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayName": "ExampleCorpGateway",
    "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayPlatform": {
        "greengrass": {
            "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"
        }
    },
    "gatewayCapabilitySummaries": [
        {
            "capabilityNamespace": "iotsitewise:opcuacollector:1",
            "capabilitySyncStatus": "IN_SYNC"
        }
    ],
    "creationDate": 1588369971.457,
    "lastUpdateDate": 1588369971.457
}
```

Pour plus d'informations, consultez la section Ingestion de données à l'aide d'une passerelle dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeGatewayà la section Référence des AWS CLI commandes.

`describe-logging-options`

L'exemple de code suivant montre comment utiliser`describe-logging-options`.

AWS CLI

Pour récupérer les options de SiteWise journalisation actuelles de AWS l'IoT

L'`describe-logging-options`exemple suivant permet de récupérer les options de SiteWise journalisation AWS IoT actuelles pour votre AWS compte dans la région actuelle.

```
aws iotsitewise describe-logging-options
```

Sortie :

```
{
    "loggingOptions": {
        "level": "INFO"
    }
}
```

Pour plus d'informations, consultez la section Surveillance de AWS l'IoT SiteWise avec Amazon CloudWatch Logs dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribeLoggingOptionsà la section Référence des AWS CLI commandes.

`describe-portal`

L'exemple de code suivant montre comment utiliser`describe-portal`.

AWS CLI

Pour décrire un portail

L'`describe-portal`exemple suivant décrit un portail Web pour une société de parc éolien.

```
aws iotsitewise describe-portal \
```

```
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Sortie :

```
{
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/
a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalName": "WindFarmPortal",
    "portalDescription": "A portal that contains wind farm projects for Example
 Corp.",
    "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",
    "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
    "portalContactEmail": "support@example.com",
    "portalStatus": {
        "state": "ACTIVE"
    },
    "portalCreationDate": "2020-02-04T23:01:52.90248068Z",
    "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",
    "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"
}
```

Pour plus d'informations, consultez la section Administration de vos portails dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DescribePortalà la section Référence des AWS CLI commandes.

`describe-project`

L'exemple de code suivant montre comment utiliser`describe-project`.

AWS CLI

Pour décrire un projet

L'`describe-project`exemple suivant décrit un projet de parc éolien.

```
aws iotsitewise describe-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

Sortie :

```
{
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectName": "Wind Farm 1",
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "projectDescription": "Contains asset visualizations for Wind Farm #1 for
 Example Corp.",
    "projectCreationDate": "2020-02-20T21:58:43.362246001Z",
    "projectLastUpdateDate": "2020-02-20T21:58:43.362246095Z"
}
```

Pour plus d'informations, consultez la section Affichage des détails du projet dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous DescribeProjectà la section Référence des AWS CLI commandes.

`disassociate-assets`

L'exemple de code suivant montre comment utiliser`disassociate-assets`.

AWS CLI

Pour dissocier un actif enfant d'un actif parent

L'`disassociate-assets`exemple suivant dissocie un actif d'éolienne d'un actif de parc éolien.

```
aws iotsitewise disassociate-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \
    --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Associer des actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DisassociateAssetsà la section Référence des AWS CLI commandes.

`get-asset-property-aggregates`

L'exemple de code suivant montre comment utiliser`get-asset-property-aggregates`.

AWS CLI

Pour récupérer les valeurs moyennes et dénombrées agrégées d'une propriété d'actif

L'`get-asset-property-aggregates`exemple suivant permet de récupérer la puissance totale moyenne d'une éolienne et le nombre de points de données de puissance totale sur une période d'une heure.

```
aws iotsitewise get-asset-property-aggregates \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --start-date 1580849400 \
    --end-date 1580853000 \
    --aggregate-types AVERAGE COUNT \
    --resolution 1h
```

Sortie :

```
{
    "aggregatedValues": [
        {
            "timestamp": 1580850000.0,
            "quality": "GOOD",
            "value": {
                "average": 8723.46538886233,
                "count": 12.0
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Interrogation des agrégats de propriétés des actifs dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetAssetPropertyAggregatesà la section Référence des AWS CLI commandes.

`get-asset-property-value-history`

L'exemple de code suivant montre comment utiliser`get-asset-property-value-history`.

AWS CLI

Pour récupérer les valeurs historiques d'une propriété d'actif

L'`get-asset-property-value-history`exemple suivant permet de récupérer les valeurs de puissance totale d'une éolienne sur une période de 20 minutes.

```
aws iotsitewise get-asset-property-value-history \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --start-date 1580851800 \
    --end-date 1580853000
```

Sortie :

```
{
    "assetPropertyValueHistory": [
        {
            "value": {
                "doubleValue": 7217.787046814844
            },
            "timestamp": {
                "timeInSeconds": 1580852100,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        },
        {
            "value": {
                "doubleValue": 6941.242811875451
            },
            "timestamp": {
                "timeInSeconds": 1580852400,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        },
        {
            "value": {
```

```
                "doubleValue": 6976.797662266717
            },
            "timestamp": {
                "timeInSeconds": 1580852700,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        },
        {
            "value": {
                "doubleValue": 6890.8677520453875
            },
            "timestamp": {
                "timeInSeconds": 1580853000,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        }
    ]
}
```

Pour plus d'informations, consultez la section Interrogation des valeurs historiques des propriétés des actifs dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetAssetPropertyValueHistoryà la section Référence des AWS CLI commandes.

`get-asset-property-value`

L'exemple de code suivant montre comment utiliser`get-asset-property-value`.

AWS CLI

Pour récupérer la valeur actuelle d'une propriété d'actif

L'`get-asset-property-value`exemple suivant permet de récupérer la puissance totale actuelle d'une éolienne.

```
aws iotsitewise get-asset-property-value \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE
```

Sortie :

```
{
    "propertyValue": {
        "value": {
            "doubleValue": 6890.8677520453875
        },
        "timestamp": {
            "timeInSeconds": 1580853000,
            "offsetInNanos": 0
        },
        "quality": "GOOD"
    }
}
```

Pour plus d'informations, consultez la section Interrogation des valeurs actuelles des propriétés des actifs dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetAssetPropertyValueà la section Référence des AWS CLI commandes.

`list-access-policies`

L'exemple de code suivant montre comment utiliser`list-access-policies`.

AWS CLI

Pour répertorier toutes les politiques d'accès

L'`list-access-policies`exemple suivant répertorie toutes les politiques d'accès pour un utilisateur administrateur du portail.

```
aws iotsitewise list-access-policies \
    --identity-type USER \
    --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE
```

Sortie :

```
{
    "accessPolicySummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
            "identity": {
```

```
                "user": {
                    "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
                }
            },
            "resource": {
                "portal": {
                    "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
                }
            },
            "permission": "ADMINISTRATOR"
        }
    ]
}
```

Pour plus d'informations, consultez la section Administration de vos portails dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListAccessPoliciesà la section Référence des AWS CLI commandes.

`list-asset-models`

L'exemple de code suivant montre comment utiliser`list-asset-models`.

AWS CLI

Pour répertorier tous les modèles d'actifs

L'`list-asset-models`exemple suivant répertorie tous les modèles d'actifs définis dans votre AWS compte dans la région actuelle.

```
aws iotsitewise list-asset-models
```

Sortie :

```
{
    "assetModelSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
```

```
            "name": "Wind Farm Model",
            "description": "Represents a wind farm that comprises many wind
 turbines",
            "creationDate": 1575671284.0,
            "lastUpdateDate": 1575671988.0,
            "status": {
                "state": "ACTIVE"
            }
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "name": "Wind Turbine Model",
            "description": "Represents a wind turbine manufactured by Example Corp",
            "creationDate": 1575671207.0,
            "lastUpdateDate": 1575686273.0,
            "status": {
                "state": "ACTIVE"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Liste de tous les modèles d'actifs dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListAssetModelsà la section Référence des AWS CLI commandes.

`list-assets`

L'exemple de code suivant montre comment utiliser`list-assets`.

AWS CLI

Exemple 1 : pour répertorier tous les actifs de premier niveau

L'`list-assets`exemple suivant répertorie tous les actifs situés au niveau supérieur de l'arborescence hiérarchique des actifs et définis dans votre AWS compte dans la région actuelle.

```
aws iotsitewise list-assets \
```

```
    --filter TOP_LEVEL
```

Sortie :

```
{
    "assetSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
            "name": "Wind Farm 1",
            "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "creationDate": 1575672453.0,
            "lastUpdateDate": 1575672453.0,
            "status": {
                "state": "ACTIVE"
            },
            "hierarchies": [
                {
                    "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
                    "name": "Wind Turbines"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

Exemple 2 : pour répertorier tous les actifs en fonction d'un modèle d'actifs

L'`list-assets`exemple suivant répertorie tous les actifs basés sur un modèle d'actif défini dans votre AWS compte dans la région actuelle.

```
aws iotsitewise list-assets \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
```

```
    "assetSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "name": "Wind Turbine 1",
            "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "creationDate": 1575671550.0,
            "lastUpdateDate": 1575686308.0,
            "status": {
                "state": "ACTIVE"
            },
            "hierarchies": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListAssetsà la section Référence des AWS CLI commandes.

`list-associated-assets`

L'exemple de code suivant montre comment utiliser`list-associated-assets`.

AWS CLI

Pour répertorier tous les actifs associés à un actif dans une hiérarchie spécifique

L'`list-associated-assets`exemple suivant répertorie tous les actifs d'éoliennes associés à l'actif de parc éolien spécifié.

```
aws iotsitewise list-associated-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE
```

Sortie :

```
{
```

```
    "assetSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "name": "Wind Turbine 1",
            "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "creationDate": 1575671550.0,
            "lastUpdateDate": 1575686308.0,
            "status": {
                "state": "ACTIVE"
            },
            "hierarchies": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Répertorier les actifs associés à un actif spécifique dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListAssociatedAssetsà la section Référence des AWS CLI commandes.

`list-dashboards`

L'exemple de code suivant montre comment utiliser`list-dashboards`.

AWS CLI

Pour répertorier tous les tableaux de bord d'un projet

L'`list-dashboards`exemple suivant répertorie tous les tableaux de bord définis dans un projet.

```
aws iotsitewise list-dashboards \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

Sortie :

```
{
    "dashboardSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
```

```
            "name": "Wind Farm",
            "creationDate": "2020-05-01T20:32:12.228476348Z",
            "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des tableaux de bord dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous ListDashboardsà la section Référence des AWS CLI commandes.

`list-gateways`

L'exemple de code suivant montre comment utiliser`list-gateways`.

AWS CLI

Pour répertorier toutes les passerelles

L'`list-gateways`exemple suivant répertorie toutes les passerelles définies dans votre AWS compte dans la région actuelle.

```
aws iotsitewise list-gateways
```

Sortie :

```
{
    "gatewaySummaries": [
        {
            "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
            "gatewayName": "ExampleCorpGateway",
            "gatewayCapabilitySummaries": [
                {
                    "capabilityNamespace": "iotsitewise:opcuacollector:1",
                    "capabilitySyncStatus": "IN_SYNC"
                }
            ],
            "creationDate": 1588369971.457,
            "lastUpdateDate": 1588369971.457
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Ingestion de données à l'aide d'une passerelle dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListGatewaysà la section Référence des AWS CLI commandes.

`list-portals`

L'exemple de code suivant montre comment utiliser`list-portals`.

AWS CLI

Pour répertorier tous les portails

L'`list-portals`exemple suivant répertorie tous les portails définis dans votre AWS compte dans la région actuelle.

```
aws iotsitewise list-portals
```

Sortie :

```
{
    "portalSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
            "name": "WindFarmPortal",
            "description": "A portal that contains wind farm projects for Example
 Corp.",
            "startUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
            "creationDate": "2020-02-04T23:01:52.90248068Z",
            "lastUpdateDate": "2020-02-04T23:01:52.90248078Z",
            "roleArn": "arn:aws:iam::123456789012:role/service-role/
MySiteWiseMonitorServiceRole"
        }
    ]
}
```

Pour plus d'informations, consultez la section Administration de vos portails dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListPortalsà la section Référence des AWS CLI commandes.

`list-project-assets`

L'exemple de code suivant montre comment utiliser`list-project-assets`.

AWS CLI

Pour répertorier tous les actifs associés à un projet

L'`list-project-assets`exemple suivant répertorie tous les actifs associés à un projet de parc éolien.

```
aws iotsitewise list-projects \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

Sortie :

```
{
    "assetIds": [
        "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
    ]
}
```

Pour plus d'informations, consultez la section Ajouter des actifs aux projets dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous ListProjectAssetsà la section Référence des AWS CLI commandes.

`list-projects`

L'exemple de code suivant montre comment utiliser`list-projects`.

AWS CLI

Pour répertorier tous les projets dans un portail

L'`list-projects`exemple suivant répertorie tous les projets définis dans un portail.

```
aws iotsitewise list-projects \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Sortie :

```
{
    "projectSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
            "name": "Wind Farm 1",
            "description": "Contains asset visualizations for Wind Farm #1 for
 Example Corp.",
            "creationDate": "2020-02-20T21:58:43.362246001Z",
            "lastUpdateDate": "2020-02-20T21:58:43.362246095Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails du projet dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous ListProjectsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier tous les tags d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'un actif d'éolienne.

```
aws iotsitewise list-tags-for-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Sortie :

```
{
    "tags": {
        "Owner": "richard-roe"
    }
}
```

Pour plus d'informations, consultez la section Marquage de vos ressources dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-logging-options`

L'exemple de code suivant montre comment utiliser`put-logging-options`.

AWS CLI

Pour spécifier le niveau de journalisation

L'`put-logging-options`exemple suivant active la journalisation des `INFO` niveaux dans AWS l'IoT SiteWise. Les autres niveaux incluent `DEBUG` et`OFF`.

```
aws iotsitewise put-logging-options \
    --logging-options level=INFO
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Surveillance de AWS l'IoT SiteWise avec Amazon CloudWatch Logs dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous PutLoggingOptionsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Ajout d’une balise à une ressource

L'`tag-resource`exemple suivant ajoute une étiquette de propriétaire à un actif d'éolienne. Cela vous permet de contrôler l'accès à l'actif en fonction de son propriétaire.

```
aws iotsitewise tag-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --tags Owner=richard-roe
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime une étiquette de propriétaire d'un actif d'éolienne.

```
aws iotsitewise untag-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --tag-keys Owner
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos ressources dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-access-policy`

L'exemple de code suivant montre comment utiliser`update-access-policy`.

AWS CLI

Pour octroyer à un observateur de projet la propriété d'un projet

L'`update-access-policy`exemple suivant met à jour une politique d'accès qui accorde à un observateur de projet la propriété d'un projet.

```
aws iotsitewise update-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \
    --cli-input-json file://update-project-viewer-access-policy.json
```

Contenu de `update-project-viewer-access-policy.json` :

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
        "project": {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
        }
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Affectation de responsables de projets dans le guide d'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous UpdateAccessPolicyà la section Référence des AWS CLI commandes.

`update-asset-model`

L'exemple de code suivant montre comment utiliser`update-asset-model`.

AWS CLI

Pour mettre à jour un modèle d'actifs

L'`update-asset-model`exemple suivant met à jour la description d'un modèle d'actifs de parc éolien. Cet exemple inclut l'existant IDs et les définitions du modèle, car il `update-asset-model` remplace le modèle existant par le nouveau modèle.

```
aws iotsitewise update-asset-model \
    --cli-input-json file://update-wind-farm-model.json
```

Contenu de `update-wind-farm-model.json` :

```
{
    "assetModelName": "Wind Farm Model",
    "assetModelDescription": "Represents a wind farm that comprises many wind
 turbines",
    "assetModelProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
            "dataType": "STRING",
            "type": {
                "attribute": {}
            }
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "metric": {
                    "expression": "sum(power)",
                    "variables": [
                        {
                            "name": "power",
                            "value": {
                                "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE",
                                "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
```

```
                        }
                    }
                }
            }
        }
    ],
    "assetModelHierarchies": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines",
            "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

Sortie :

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour des modèles d'actifs dans le Guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateAssetModelà la section Référence des AWS CLI commandes.

`update-asset-property`

L'exemple de code suivant montre comment utiliser`update-asset-property`.

AWS CLI

Exemple 1 : pour mettre à jour l'alias d'une propriété d'actif

L'`update-asset-property`exemple suivant met à jour l'alias de propriété énergétique d'un actif d'éolienne.

```
aws iotsitewise update-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-55555EXAMPLE \
    --property-alias "/examplecorp/windfarm/1/turbine/1/power" \
    --property-notification-state DISABLED
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Mappage des flux de données industriels aux propriétés des actifs dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

Exemple 2 : pour activer les notifications relatives aux propriétés des actifs

L'`update-asset-property`exemple suivant active les notifications de mise à jour des propriétés énergétiques d'une éolienne. Les mises à jour de la valeur des propriétés sont publiées dans la rubrique MQTT`$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>`, où chaque ID est remplacé par l'ID de propriété, d'actif et de modèle de la propriété de l'actif.

```
aws iotsitewise update-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --property-notification-state ENABLED \
    --property-alias "/examplecorp/windfarm/1/turbine/1/power"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Interaction avec d'autres services dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateAssetPropertyà la section Référence des AWS CLI commandes.

`update-asset`

L'exemple de code suivant montre comment utiliser`update-asset`.

AWS CLI

Pour mettre à jour le nom d'un actif

L'`update-asset`exemple suivant met à jour le nom d'un actif d'éolienne.

```
aws iotsitewise update-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --asset-name "Wind Turbine 2"
```

Sortie :

```
{
    "assetStatus": {
        "state": "UPDATING"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour des actifs dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateAssetà la section Référence des AWS CLI commandes.

`update-dashboard`

L'exemple de code suivant montre comment utiliser`update-dashboard`.

AWS CLI

Pour mettre à jour un tableau de bord

L'`update-dashboard`exemple suivant modifie le titre du graphique linéaire d'un tableau de bord qui affiche la puissance totale produite pour un parc éolien.

```
aws iotsitewise update-dashboard \
    --project-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE \
    --dashboard-name "Wind Farm" \
    --dashboard-definition file://update-wind-farm-dashboard.json
```

Contenu de `update-wind-farm-dashboard.json` :

```
{
    "widgets": [
```

```
        {
            "type": "monitor-line-chart",
            "title": "Total Generated Power",
            "x": 0,
            "y": 0,
            "height": 3,
            "width": 3,
            "metrics": [
                {
                    "label": "Power",
                    "type": "iotsitewise",
                    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
                    "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"
                }
            ]
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création de tableaux de bord (CLI) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateDashboardà la section Référence des AWS CLI commandes.

`update-gateway-capability-configuration`

L'exemple de code suivant montre comment utiliser`update-gateway-capability-configuration`.

AWS CLI

Pour mettre à jour les fonctionnalités d'une passerelle

L'`update-gateway-capability-configuration`exemple suivant configure une source OPC-UA avec les propriétés suivantes :

Fait confiance à n'importe quel certificat. Utilise l'algorithme Basic256 pour sécuriser les messages.Utilise le SignAndEncrypt mode pour sécuriser les connexions. Utilise les informations d'authentification stockées dans un secret de Secrets Manager. AWS

```
aws iotsitewise update-gateway-capability-configuration \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --capability-namespace "iotsitewise:opcuacollector:1" \
    --capability-configuration file://opc-ua-capability-configuration.json
```

Contenu de `opc-ua-capability-configuration.json` :

```
{
    "sources": [
        {
            "name": "Wind Farm #1",
            "endpoint": {
                "certificateTrust": {
                    "type": "TrustAny"
                },
                "endpointUri": "opc.tcp://203.0.113.0:49320",
                "securityPolicy": "BASIC256",
                "messageSecurityMode": "SIGN_AND_ENCRYPT",
                "identityProvider": {
                    "type": "Username",
                    "usernameSecretArn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-windfarm1-auth-1ABCDE"
                },
                "nodeFilterRules": []
            },
            "measurementDataStreamPrefix": ""
        }
    ]
}
```

Sortie :

```
{
    "capabilityNamespace": "iotsitewise:opcuacollector:1",
    "capabilitySyncStatus": "OUT_OF_SYNC"
}
```

Pour plus d'informations, consultez la section Configuration des sources de données dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateGatewayCapabilityConfigurationà la section Référence des AWS CLI commandes.

update-gateway

L'exemple de code suivant montre comment utiliserupdate-gateway.

AWS CLI

Pour mettre à jour le nom d'une passerelle

L'update-gatewayexemple suivant met à jour le nom d'une passerelle.

```
aws iotsitewise update-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --gateway-name ExampleCorpGateway1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ingestion de données à l'aide d'une passerelle dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateGatewayà la section Référence des AWS CLI commandes.

update-portal

L'exemple de code suivant montre comment utiliserupdate-portal.

AWS CLI

Pour mettre à jour les informations d'un portail

L'update-portalexemple suivant met à jour un portail Web pour une société de parc éolien.

```
aws iotsitewise update-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \
    --portal-name WindFarmPortal \
    --portal-description "A portal that contains wind farm projects for Example
 Corp." \
    --portal-contact-email support@example.com \
    --role-arn arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole
```

Sortie :

```
{
    "portalStatus": {
        "state": "UPDATING"
    }
}
```

Pour plus d'informations, consultez la section Administration de vos portails dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdatePortalà la section Référence des AWS CLI commandes.

`update-project`

L'exemple de code suivant montre comment utiliser`update-project`.

AWS CLI

Pour mettre à jour les détails d'un projet

L'`update-project`exemple suivant met à jour un projet de parc éolien.

```
aws iotsitewise update-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --project-name "Wind Farm 1" \
    --project-description "Contains asset visualizations for Wind Farm #1 for
 Example Corp."
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modification des détails du projet dans le guide de l'application AWS IoT SiteWise Monitor.

- Pour plus de détails sur l'API, reportez-vous UpdateProjectà la section Référence des AWS CLI commandes.

AWS IoT Things Graph exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Things Graph.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-entity-to-thing`

L'exemple de code suivant montre comment utiliser`associate-entity-to-thing`.

AWS CLI

Pour associer un objet à un appareil

L'`associate-entity-to-thing`exemple suivant associe un objet à un appareil. L'exemple utilise un dispositif de détection de mouvement qui se trouve dans l'espace de noms public.

```
aws iotthingsgraph associate-entity-to-thing \
    --thing-name "MotionSensorName" \
    --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création et téléchargement de modèles dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous AssociateEntityToThingà la section Référence des AWS CLI commandes.

`create-flow-template`

L'exemple de code suivant montre comment utiliser`create-flow-template`.

AWS CLI

Pour créer un flux

L'`create-flow-template`exemple suivant crée un flux (flux de travail). La valeur de `MyFlowDefinition` est le GraphQL qui modélise le flux.

```
aws iotthingsgraph create-flow-template \
    --definition language=GRAPHQL,text="MyFlowDefinition"
```

Sortie :

```
{
    "summary": {
        "createdAt": 1559248067.545,
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 1
    }
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous CreateFlowTemplateà la section Référence des AWS CLI commandes.

`create-system-instance`

L'exemple de code suivant montre comment utiliser`create-system-instance`.

AWS CLI

Pour créer une instance de système

L'`create-system-instance`exemple suivant crée une instance de système. La valeur de `MySystemInstanceDefinition` est le GraphQL qui modélise l'instance du système.

```
aws iotthingsgraph create-system-instance -\
    -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
    --target CLOUD \
    --flow-actions-role-arn myRoleARN
```

Sortie :

```
{
```

```
    "summary": {
        "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/
Room218",
        "status": "NOT_DEPLOYED",
        "target": "CLOUD",
        "createdAt": 1559249315.208,
        "updatedAt": 1559249315.208
    }
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous CreateSystemInstanceà la section Référence des AWS CLI commandes.

`create-system-template`

L'exemple de code suivant montre comment utiliser`create-system-template`.

AWS CLI

Pour créer un système

L'`create-system-template`exemple suivant crée un système. La valeur de MySystemDefinition est le GraphQL qui modélise le système.

```
aws iotthingsgraph create-system-template \
    --definition language=GRAPHQL,text="MySystemDefinition"
```

Sortie :

```
{
    "summary": {
        "createdAt": 1559249776.254,
        "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MySystem",
        "revisionNumber": 1
    }
}
```

Pour plus d'informations, consultez la section Création de systèmes dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous CreateSystemTemplateà la section Référence des AWS CLI commandes.

`delete-flow-template`

L'exemple de code suivant montre comment utiliser`delete-flow-template`.

AWS CLI

Pour supprimer un flux

L'`delete-flow-template`exemple suivant supprime un flux (flux de travail).

```
aws iotthingsgraph delete-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeleteFlowTemplateà la section Référence des AWS CLI commandes.

`delete-namespace`

L'exemple de code suivant montre comment utiliser`delete-namespace`.

AWS CLI

Pour supprimer un espace de noms

L'`delete-namespace`exemple suivant supprime un espace de noms.

```
aws iotthingsgraph delete-namespace
```

Sortie :

```
{
   "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
   "namespaceName": "us-west-2/123456789012/default"
}
```

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeleteNamespaceà la section Référence des AWS CLI commandes.

`delete-system-instance`

L'exemple de code suivant montre comment utiliser`delete-system-instance`.

AWS CLI

Pour supprimer une instance du système

L'`delete-system-instance`exemple suivant supprime une instance du système.

```
aws iotthingsgraph delete-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeleteSystemInstanceà la section Référence des AWS CLI commandes.

`delete-system-template`

L'exemple de code suivant montre comment utiliser`delete-system-template`.

AWS CLI

Pour supprimer un système

L'`delete-system-template`exemple suivant supprime un système.

```
aws iotthingsgraph delete-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeleteSystemTemplateà la section Référence des AWS CLI commandes.

`deploy-system-instance`

L'exemple de code suivant montre comment utiliser`deploy-system-instance`.

AWS CLI

Pour déployer une instance de système

L'`delete-system-template`exemple suivant déploie une instance de système.

```
aws iotthingsgraph deploy-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Sortie :

```
{
   "summary": {
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment:Room218",
      "createdAt": 1559249776.254,
      "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
      "status": "DEPLOYED_IN_TARGET",
      "target": "CLOUD",
      "updatedAt": 1559249776.254
   }
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeploySystemInstanceà la section Référence des AWS CLI commandes.

`deprecate-flow-template`

L'exemple de code suivant montre comment utiliser`deprecate-flow-template`.

AWS CLI

Pour déprécier un flux

L'`deprecate-flow-template`exemple suivant déconseille un flux (flux de travail).

```
aws iotthingsgraph deprecate-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeprecateFlowTemplateà la section Référence des AWS CLI commandes.

`deprecate-system-template`

L'exemple de code suivant montre comment utiliser`deprecate-system-template`.

AWS CLI

Pour déprécier un système

L'`deprecate-system-template`exemple suivant déconseille un système.

```
aws iotthingsgraph deprecate-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DeprecateSystemTemplateà la section Référence des AWS CLI commandes.

`describe-namespace`

L'exemple de code suivant montre comment utiliser`describe-namespace`.

AWS CLI

Pour obtenir une description de votre espace de noms

L'`describe-namespace`exemple suivant permet d'obtenir une description de votre espace de noms.

```
aws iotthingsgraph describe-namespace
```

Sortie :

```
{
    "namespaceName": "us-west-2/123456789012/default",
    "trackingNamespaceName": "aws",
    "trackingNamespaceVersion": 1,
    "namespaceVersion": 5
}
```

Pour plus d'informations, consultez la section Namespaces dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DescribeNamespaceà la section Référence des AWS CLI commandes.

`dissociate-entity-from-thing`

L'exemple de code suivant montre comment utiliser`dissociate-entity-from-thing`.

AWS CLI

Pour dissocier un objet d'un appareil

L'`dissociate-entity-from-thing`exemple suivant dissocie un objet d'un appareil.

```
aws iotthingsgraph dissociate-entity-from-thing \
    --thing-name "MotionSensorName" \
    --entity-type "DEVICE"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création et téléchargement de modèles dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous DissociateEntityFromThingà la section Référence des AWS CLI commandes.

`get-entities`

L'exemple de code suivant montre comment utiliser`get-entities`.

AWS CLI

Pour obtenir des définitions pour des entités

L'`get-entities`exemple suivant permet d'obtenir la définition d'un modèle d'appareil.

```
aws iotthingsgraph get-entities \
    --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

Sortie :

```
{
    "descriptions": [
        {
            "id": "urn:tdm:aws/examples:DeviceModel:MotionSensor",
            "type": "DEVICE_MODEL",
            "createdAt": 1559256190.599,
            "definition": {
                "language": "GRAPHQL",
                "text": "##\n# Specification of motion sensor devices interface.\n##
\ntype MotionSensor @deviceModel(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\",\n        capability: \"urn:tdm:aws/examples:capability:MotionSensorCapability\")
 {ignore:void}"
            }
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Création et téléchargement de modèles dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetEntitiesà la section Référence des AWS CLI commandes.

`get-flow-template-revisions`

L'exemple de code suivant montre comment utiliser`get-flow-template-revisions`.

AWS CLI

Pour obtenir des informations de révision concernant un flux

L'`get-flow-template-revisions`exemple suivant permet d'obtenir des informations de révision concernant un flux (flux de travail).

```
aws iotthingsgraph get-flow-template-revisions \
    --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow
```

Sortie :

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
        }
    ]
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetFlowTemplateRevisionsà la section Référence des AWS CLI commandes.

`get-flow-template`

L'exemple de code suivant montre comment utiliser`get-flow-template`.

AWS CLI

Pour obtenir une définition de flux

L'`get-flow-template`exemple suivant obtient la définition d'un flux (flux de travail).

```
aws iotthingsgraph get-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Sortie :

```
{
    "description": {
        "summary": {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
        },
        "definition": {
            "language": "GRAPHQL",
            "text": "{\nquery MyFlow($camera: string!, $screen: string!)
 @workflowType(id: \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\")
 @annotation(type: \"tgc:FlowEvent\", id: \"sledged790c1b2bcd949e09da0c9bfc077f79d
\", x: 1586, y: 653) @triggers(definition: \"{MotionSensor(description:
 \\\"\\\") @position(x: 1045, y: 635.6666564941406) {\\n  condition(expr:
 \\\"devices[name == \\\\\\\"motionSensor\\\\\\\"].events[name == \\\
\\\"StateChanged\\\\\\\"].lastEvent\\\")\\n  action(expr: \\\"\\\")\
\n}}\") {\n  variables {\n    cameraResult @property(id: \"urn:tdm:aws/
examples:property:CameraStateProperty\")\n  }\n  steps {\n    step(name: \"Camera
\", outEvent: [\"sledged790c1b2bcd949e09da0c9bfc077f79d\"]) @position(x: 1377,
 y: 638.6666564941406) {\n      DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Camera\", out: \"cameraResult\", deviceId: \"${camera}\")
 {\n        capture\n      }\n    }\n    step(name: \"Screen\", inEvent:
 [\"sledged790c1b2bcd949e09da0c9bfc077f79d\"]) @position(x: 1675.6666870117188,
 y: 637.9999847412109) {\n      DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Screen\", deviceId: \"${screen}\") {\n        display(imageUrl:
 \"${cameraResult.lastClickedImage}\")\n      }\n    }\n  }\n}\n}"
        },
        "validatedNamespaceVersion": 5
    }
```

```
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetFlowTemplateà la section Référence des AWS CLI commandes.

`get-namespace-deletion-status`

L'exemple de code suivant montre comment utiliser`get-namespace-deletion-status`.

AWS CLI

Pour obtenir le statut de la tâche de suppression de l'espace de noms

L'`get-namespace-deletion-status`exemple suivant permet d'obtenir le statut de la tâche de suppression de l'espace de noms.

```
aws iotthingsgraph get-namespace-deletion-status
```

Sortie :

```
{
   "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
   "namespaceName": "us-west-2/123456789012/default"
   "status": "SUCCEEDED "
}
```

Pour plus d'informations, consultez la section Namespaces dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetNamespaceDeletionStatusà la section Référence des AWS CLI commandes.

`get-system-instance`

L'exemple de code suivant montre comment utiliser`get-system-instance`.

AWS CLI

Pour obtenir une instance de système

L'`get-system-instance`exemple suivant permet d'obtenir une définition pour une instance de système.

```
aws iotthingsgraph get-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Sortie :

```
{
    "description": {
        "summary": {
            "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218",
            "status": "NOT_DEPLOYED",
            "target": "CLOUD",
            "createdAt": 1559249315.208,
            "updatedAt": 1559249315.208
        },
        "definition": {
            "language": "GRAPHQL",
            "text": "{\r\nquery Room218 @deployment(id: \"urn:tdm:us-
west-2/123456789012/default:Deployment:Room218\", systemId: \"urn:tdm:us-
west-2/123456789012/default:System:SecurityFlow\") {\r\n    motionSensor(deviceId:
 \"MotionSensorName\")\r\n    screen(deviceId: \"ScreenName\")\r\n
 camera(deviceId: \"CameraName\") \r\n    triggers {MotionEventTrigger(description:
 \"a trigger\") {  \r\n    condition(expr: \"devices[name ==
 'motionSensor'].events[name == 'StateChanged'].lastEvent\") \r\n    action(expr:
 \"ThingsGraph.startFlow('SecurityFlow', bindings[name == 'camera'].deviceId,
 bindings[name == 'screen'].deviceId)\")\r\n    }\r\n   }\r\n  }\r\n  }"
        },
        "metricsConfiguration": {
            "cloudMetricEnabled": false
        },
        "validatedNamespaceVersion": 5,
        "flowActionsRoleArn": "arn:aws:iam::123456789012:role/ThingsGraphRole"
    }
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetSystemInstanceà la section Référence des AWS CLI commandes.

`get-system-template-revisions`

L'exemple de code suivant montre comment utiliser`get-system-template-revisions`.

AWS CLI

Pour obtenir des informations de révision concernant un système

L'`get-system-template-revisions`exemple suivant permet d'obtenir des informations de révision concernant un système.

```
aws iotthingsgraph get-system-template-revisions \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Sortie :

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MySystem",
            "revisionNumber": 1,
            "createdAt": 1559247540.656
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetSystemTemplateRevisionsà la section Référence des AWS CLI commandes.

`get-system-template`

L'exemple de code suivant montre comment utiliser`get-system-template`.

AWS CLI

Pour obtenir un système

L'`get-system-template`exemple suivant permet d'obtenir la définition d'un système.

```
aws iotthingsgraph get-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Sortie :

```
{
    "description": {
        "summary": {
            "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.656
        },
        "definition": {
            "language": "GRAPHQL",
            "text": "{\ntype MySystem @systemType(id: \"urn:tdm:us-
west-2/123456789012/default:System:MySystem\", description: \"\") {\n  camera:
 Camera @thing(id: \"urn:tdm:aws/examples:deviceModel:Camera\")\n  screen:
 Screen @thing(id: \"urn:tdm:aws/examples:deviceModel:Screen\")\n  motionSensor:
 MotionSensor @thing(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\")\n  MyFlow: MyFlow @workflow(id: \"urn:tdm:us-west-2/123456789012/
default:Workflow:MyFlow\")\n}\n}"
        },
        "validatedNamespaceVersion": 5
    }
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetSystemTemplateà la section Référence des AWS CLI commandes.

`get-upload-status`

L'exemple de code suivant montre comment utiliser`get-upload-status`.

AWS CLI

Pour connaître le statut de votre entité, téléchargez

L'`get-upload-status`exemple suivant permet d'obtenir le statut de votre opération de téléchargement d'entités. La valeur de `MyUploadId` est la valeur d'ID renvoyée par l'`upload-entity-definitions`opération.

```
aws iotthingsgraph get-upload-status \
    --upload-id "MyUploadId"
```

Sortie :

```
{
    "namespaceName": "us-west-2/123456789012/default",
    "namespaceVersion": 5,
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
    "uploadStatus": "SUCCEEDED"
}
```

Pour plus d'informations, consultez la section Modeling Entities dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous GetUploadStatusà la section Référence des AWS CLI commandes.

`list-flow-execution-messages`

L'exemple de code suivant montre comment utiliser`list-flow-execution-messages`.

AWS CLI

Pour obtenir des informations sur les événements liés à l'exécution d'un flux

L'`list-flow-execution-messages`exemple suivant permet d'obtenir des informations sur les événements d'une exécution de flux.

```
aws iotthingsgraph list-flow-execution-messages \
    --flow-execution-id "urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow_2019-05-11T19:39:55.317Z_MotionSensor_69b151ad-
a611-42f5-ac21-fe537f9868ad"
```

Sortie :

```
{
    "messages": [
        {
         "eventType": "EXECUTION_STARTED",
         "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
         "payload": "Flow execution started",
         "timestamp": 1559247540.656
        }
    ]
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous ListFlowExecutionMessagesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier tous les tags d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'une ressource AWS IoT Things Graph.

```
aws iotthingsgraph list-tags-for-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218"
```

Sortie :

```
{
   "tags": [
      {
         "key": "Type",
         "value": "Residential"
      }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Tagging Your AWS IoT Things Graph Resources dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`search-entities`

L'exemple de code suivant montre comment utiliser`search-entities`.

AWS CLI

Pour rechercher des entités

L'`search-entities`exemple suivant recherche toutes les entités de typeEVENT.

```
aws iotthingsgraph search-entities \
    --entity-types "EVENT"
```

Sortie :

```
{
    "descriptions": [
        {
            "id": "urn:tdm:aws/examples:Event:MotionSensorEvent",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
                "text": "##\n# Description of events emitted by motion
 sensor.\n##\ntype MotionSensorEvent @eventType(id: \"urn:tdm:aws/
examples:event:MotionSensorEvent\",\n            payload: \"urn:tdm:aws/
examples:property:MotionSensorStateProperty\") {ignore:void}"
            }
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Event:CameraClickedEventV2",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
```

```
                "text": "type CameraClickedEventV2 @eventType(id: \"urn:tdm:us-
west-2/123456789012/default:event:CameraClickedEventV2\",\r\npayload:
 \"urn:tdm:aws:Property:Boolean\"){ignore:void}"
            }
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Event:MotionSensorEventV2",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
                "text": "# Event emitted by the motion sensor.\r\ntype
 MotionSensorEventV2 @eventType(id: \"urn:tdm:us-west-2/123456789012/
default:event:MotionSensorEventV2\",\r\npayload: \"urn:tdm:us-west-2/123456789012/
default:property:MotionSensorStateProperty2\") {ignore:void}"
            }
        }
    ],
    "nextToken": "urn:tdm:us-west-2/123456789012/default:Event:MotionSensorEventV2"
}
```

Pour plus d'informations, consultez la référence du modèle de données AWS IoT Things Graph dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous SearchEntitiesà la section Référence des AWS CLI commandes.

`search-flow-executions`

L'exemple de code suivant montre comment utiliser`search-flow-executions`.

AWS CLI

Pour rechercher des exécutions de flux

L'`search-flow-executions`exemple suivant permet de rechercher toutes les exécutions d'un flux dans une instance système spécifiée.

```
aws iotthingsgraph search-flow-executions \
    --system-instance-id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Sortie :

```
{
   "summaries": [
      {
         "createdAt": 1559247540.656,
         "flowExecutionId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
         "flowTemplateId": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
         "status": "RUNNING ",
         "systemInstanceId": "urn:tdm:us-west-2/123456789012/
default:System:MySystem",
         "updatedAt": 1559247540.656
      }
   ]
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous SearchFlowExecutionsà la section Référence des AWS CLI commandes.

`search-flow-templates`

L'exemple de code suivant montre comment utiliser`search-flow-templates`.

AWS CLI

Pour rechercher des flux (ou des flux de travail)

L'`search-flow-templates`exemple suivant recherche tous les flux (flux de travail) contenant le modèle d'appareil photo.

```
aws iotthingsgraph search-flow-templates \
    --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

Sortie :

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
```

```
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:SecurityFlow",
            "revisionNumber": 3,
            "createdAt": 1548283099.27
        }
    ]
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous SearchFlowTemplatesà la section Référence des AWS CLI commandes.

`search-system-instances`

L'exemple de code suivant montre comment utiliser`search-system-instances`.

AWS CLI

Pour rechercher des instances du système

L'`search-system-instances`exemple suivant recherche toutes les instances du système qui contiennent le système spécifié.

```
aws iotthingsgraph search-system-instances \
    --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:System:SecurityFlow"
```

Sortie :

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:DeploymentForSample",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/DeploymentForSample",
            "status": "NOT_DEPLOYED",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGrnGr",
            "createdAt": 1555716314.707,
```

```
            "updatedAt": 1555716314.707
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment",
            "status": "DELETED_IN_TARGET",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGrnGr",
            "createdAt": 1549416462.049,
            "updatedAt": 1549416722.361,
            "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
            "greengrassGroupVersionId": "7365aed7-2d3e-4d13-aad8-75443d45eb05"
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment2",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment2",
            "status": "DEPLOYED_IN_TARGET",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGrnGr",
            "createdAt": 1549572385.774,
            "updatedAt": 1549572418.408,
            "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
            "greengrassGroupVersionId": "bfa70ab3-2bf7-409c-a4d4-bc8328ae5b86"
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room215",
            "status": "NOT_DEPLOYED",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGG",
            "createdAt": 1547056918.413,
            "updatedAt": 1547056918.413
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218",
            "status": "NOT_DEPLOYED",
            "target": "CLOUD",
```

```
            "createdAt": 1559249315.208,
            "updatedAt": 1559249315.208
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des systèmes et des configurations de flux dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous SearchSystemInstancesà la section Référence des AWS CLI commandes.

`search-system-templates`

L'exemple de code suivant montre comment utiliser`search-system-templates`.

AWS CLI

Pour rechercher un système

L'`search-system-templates`exemple suivant recherche tous les systèmes contenant le flux spécifié.

```
aws iotthingsgraph search-system-templates \
    --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow"
```

Sortie :

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:System:SecurityFlow",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
SecurityFlow",
            "revisionNumber": 1,
            "createdAt": 1548283099.433
        }
    ]
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous SearchSystemTemplatesà la section Référence des AWS CLI commandes.

`search-things`

L'exemple de code suivant montre comment utiliser`search-things`.

AWS CLI

Pour rechercher des éléments associés à des appareils et à des modèles d'appareils

L'`search-things`exemple suivant recherche tous les éléments associés au MotionSensor périphérique HCSR5 01.

```
aws iotthingsgraph search-things \
    --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Sortie :

```
{
    "things": [
        {
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MotionSensor1",
            "thingName": "MotionSensor1"
        },
        {
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/TG_MS",
            "thingName": "TG_MS"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création et téléchargement de modèles dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous SearchThingsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour créer un tag pour une ressource

L'`tag-resource`exemple suivant crée une balise pour la ressource spécifiée.

```
aws iotthingsgraph tag-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218" \
    --tags key="Type",value="Residential"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Tagging Your AWS IoT Things Graph Resources dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`undeploy-system-instance`

L'exemple de code suivant montre comment utiliser`undeploy-system-instance`.

AWS CLI

Pour annuler le déploiement d'une instance système depuis sa cible

L'`undeploy-system-instance`exemple suivant supprime une instance de système de sa cible.

```
aws iotthingsgraph undeploy-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

Sortie :

```
{
    "summary": {
```

```
        "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/
Room215",
        "status": "PENDING_DELETE",
        "target": "GREENGRASS",
        "greengrassGroupName": "ThingsGraphGrnGr",
        "createdAt": 1553189694.255,
        "updatedAt": 1559344549.601,
        "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
        "greengrassGroupVersionId": "731b371d-d644-4b67-ac64-3934e99b75d7"
    }
}
```

Pour plus d'informations, consultez la section Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments dans le Guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous UndeploySystemInstanceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer le tag d'une ressource

L'`untag-resource`exemple suivant supprime une balise pour la ressource spécifiée.

```
aws iotthingsgraph untag-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218" \
    --tag-keys "Type"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Tagging Your AWS IoT Things Graph Resources dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-flow-template`

L'exemple de code suivant montre comment utiliser`update-flow-template`.

AWS CLI

Pour mettre à jour un flux

L'`update-flow-template`exemple suivant met à jour un flux (flux de travail). La valeur de `MyFlowDefinition` est le GraphQL qui modélise le flux.

```
aws iotthingsgraph update-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \
    --definition language=GRAPHQL,text="MyFlowDefinition"
```

Sortie :

```
{
    "summary": {
        "createdAt": 1559248067.545,
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 2
    }
}
```

Pour plus d'informations, voir Working with Flows dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous UpdateFlowTemplateà la section Référence des AWS CLI commandes.

`update-system-template`

L'exemple de code suivant montre comment utiliser`update-system-template`.

AWS CLI

Pour mettre à jour un système

L'`update-system-template`exemple suivant met à jour un système. La valeur de `MySystemDefinition` est le GraphQL qui modélise le système.

```
aws iotthingsgraph update-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \
    --definition language=GRAPHQL,text="MySystemDefinition"
```

Sortie :

```
{
    "summary": {
        "createdAt": 1559249776.254,
        "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MySystem",
        "revisionNumber": 2
    }
}
```

Pour plus d'informations, consultez la section Création de systèmes dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous UpdateSystemTemplateà la section Référence des AWS CLI commandes.

`upload-entity-definitions`

L'exemple de code suivant montre comment utiliser`upload-entity-definitions`.

AWS CLI

Pour télécharger des définitions d'entités

L'`upload-entity-definitions`exemple suivant télécharge les définitions d'entités dans votre espace de noms. La valeur de `MyEntityDefinitions` est le GraphQL qui modélise les entités.

```
aws iotthingsgraph upload-entity-definitions \
    --document language=GRAPHQL,text="MyEntityDefinitions"
```

Sortie :

```
{
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"
}
```

Pour plus d'informations, consultez la section Modeling Entities dans le guide de l'utilisateur de l'AWS IoT Things Graph.

- Pour plus de détails sur l'API, reportez-vous UploadEntityDefinitionsà la section Référence des AWS CLI commandes.

AWS IoT Wireless exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS IoT Wireless.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-aws-account-with-partner-account`

L'exemple de code suivant montre comment utiliser`associate-aws-account-with-partner-account`.

AWS CLI

Pour associer un compte partenaire à votre AWS compte

L'`associate-aws-account-with-partner-account`exemple suivant associe les informations d'identification du compte Sidewalk suivantes à votre AWS compte.

```
aws iotwireless associate-aws-account-with-partner-account \
    --sidewalk
 AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78
```

Sortie :

```
{
    "Sidewalk": {
        "AmazonId": "12345678901234",
        "AppServerPrivateKey":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
    }
}
```

Pour plus d'informations, consultez Amazon Sidewalk Integration for AWS IoT Core dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir AssociateAwsAccountWithPartnerAccountla section Référence des AWS CLI commandes.

`associate-wireless-device-with-thing`

L'exemple de code suivant montre comment utiliser`associate-wireless-device-with-thing`.

AWS CLI

Pour associer un objet à un appareil sans fil

L'`associate-wireless-device-with-thing`exemple suivant associe à votre appareil sans fil un objet portant l'ID spécifié.

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir AssociateWirelessDeviceWithThingla section Référence des AWS CLI commandes.

`associate-wireless-gateway-with-certificate`

L'exemple de code suivant montre comment utiliser`associate-wireless-gateway-with-certificate`.

AWS CLI

Pour associer le certificat à la passerelle sans fil

Ce qui suit `associate-wireless-gateway-with-certificate` associe une passerelle sans fil à un certificat.

```
aws iotwireless associate-wireless-gateway-with-certificate \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --iot-certificate-
id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

Sortie :

```
{
    "IotCertificateId":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
}
```

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir AssociateWirelessGatewayWithCertificatela section Référence des AWS CLI commandes.

`associate-wireless-gateway-with-thing`

L'exemple de code suivant montre comment utiliser`associate-wireless-gateway-with-thing`.

AWS CLI

Pour associer un objet à une passerelle sans fil

L'`associate-wireless-gateway-with-thing`exemple suivant associe un objet à une passerelle sans fil.

```
aws iotwireless associate-wireless-gateway-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir AssociateWirelessGatewayWithThingla section Référence des AWS CLI commandes.

`create-destination`

L'exemple de code suivant montre comment utiliser`create-destination`.

AWS CLI

Pour créer une destination sans fil IoT

L'`create-destination`exemple suivant crée une destination pour mapper un message d'appareil à une règle AWS IoT. Avant d'exécuter cette commande, vous devez avoir créé un rôle IAM qui donne à AWS IoT Core for LoRa WAN les autorisations nécessaires pour envoyer des données à la règle AWS IoT.

```
aws iotwireless create-destination \
    --name IoTWirelessDestination \
    --expression-type RuleName \
    --expression IoTWirelessRule \
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "IoTWirelessDestination"
}
```

Pour plus d'informations, consultez la section Ajouter des destinations à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir CreateDestinationla section Référence des AWS CLI commandes.

`create-device-profile`

L'exemple de code suivant montre comment utiliser`create-device-profile`.

AWS CLI

Pour créer un nouveau profil d'appareil

L'`create-device-profile`exemple suivant crée un nouveau profil d'appareil sans fil IoT.

```
aws iotwireless create-device-profile
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir CreateDeviceProfilela section Référence des AWS CLI commandes.

`create-service-profile`

L'exemple de code suivant montre comment utiliser`create-service-profile`.

AWS CLI

Pour créer un nouveau profil de service

L'`create-service-profile`exemple suivant crée un nouveau profil de service sans fil IoT.

```
aws iotwireless create-service-profile
```

Sortie :

```
{
```

```
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir CreateServiceProfilela section Référence des AWS CLI commandes.

`create-wireless-device`

L'exemple de code suivant montre comment utiliser`create-wireless-device`.

AWS CLI

Pour créer un appareil sans fil IoT

L'`create-wireless-device`exemple suivant crée une ressource de périphérique sans fil du type LoRa WAN.

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
```

```
    "Type": LoRaWAN
}
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir CreateWirelessDevicela section Référence des AWS CLI commandes.

`create-wireless-gateway-task-definition`

L'exemple de code suivant montre comment utiliser`create-wireless-gateway-task-definition`.

AWS CLI

Pour créer une définition de tâche de passerelle sans fil

Ce qui suit crée `create-wireless-gateway-task-definition` automatiquement des tâches à l'aide de cette définition de tâche pour toutes les passerelles avec la version actuelle spécifiée.

```
aws iotwireless create-wireless-gateway-task-definition \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "AutoCreateTasks": true,
    "Name": "TestAutoUpdate",
    "Update":{
        "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",
```

```
        "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",
        "LoRaWAN" :{
            "CurrentVersion" :{
                "PackageVersion" : "1.0.0",
                "Station" : "2.0.5",
                "Model" : "linux"
            },
            "UpdateVersion" :{
                "PackageVersion" : "1.0.1",
                "Station" : "2.0.5",
                "Model" : "minihub"
            }
        }
    }
}
```

Sortie :

```
{
    "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, voir CreateWirelessGatewayTaskDefinitionla section Référence des AWS CLI commandes.

`create-wireless-gateway-task`

L'exemple de code suivant montre comment utiliser`create-wireless-gateway-task`.

AWS CLI

Pour créer la tâche pour une passerelle sans fil

L'`create-wireless-gateway-task`exemple suivant crée une tâche pour une passerelle sans fil.

```
aws iotwireless create-wireless-gateway-task \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
```

```
    --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"
```

Sortie :

```
{
    "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",
    "Status": "Success"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateWirelessGatewayTaskà la section Référence des AWS CLI commandes.

`create-wireless-gateway`

L'exemple de code suivant montre comment utiliser`create-wireless-gateway`.

AWS CLI

Pour créer une passerelle sans fil

L'`create-wireless-gateway`exemple suivant crée une passerelle de périphérique LoRa WAN sans fil.

```
aws iotwireless create-wireless-gateway \
    --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
    --name "myFirstLoRaWANGateway" \
    --description "Using my first LoRaWAN gateway"
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous CreateWirelessGatewayà la section Référence des AWS CLI commandes.

`delete-destination`

L'exemple de code suivant montre comment utiliser`delete-destination`.

AWS CLI

Pour supprimer une destination sans fil IoT

L'`delete-destination`exemple suivant supprime la ressource de destination sans fil portant le nom `IoTWirelessDestination` que vous avez créé.

```
aws iotwireless delete-destination \
    --name "IoTWirelessDestination"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des destinations à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteDestinationà la section Référence des AWS CLI commandes.

`delete-device-profile`

L'exemple de code suivant montre comment utiliser`delete-device-profile`.

AWS CLI

Pour supprimer un profil d'appareil

L'`delete-device-profile`exemple suivant supprime un profil d'appareil avec l'ID spécifié que vous avez créé.

```
aws iotwireless delete-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteDeviceProfileà la section Référence des AWS CLI commandes.

`delete-service-profile`

L'exemple de code suivant montre comment utiliser`delete-service-profile`.

AWS CLI

Pour supprimer un profil de service

L'`delete-service-profile`exemple suivant supprime un profil de service avec l'ID spécifié que vous avez créé.

```
aws iotwireless delete-service-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceProfileà la section Référence des AWS CLI commandes.

`delete-wireless-device`

L'exemple de code suivant montre comment utiliser`delete-wireless-device`.

AWS CLI

Pour supprimer un appareil sans fil

L'`delete-wireless-device`exemple suivant supprime un périphérique sans fil portant l'ID spécifié.

```
aws iotwireless delete-wireless-device \
```

```
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteWirelessDeviceà la section Référence des AWS CLI commandes.

`delete-wireless-gateway-task-definition`

L'exemple de code suivant montre comment utiliser`delete-wireless-gateway-task-definition`.

AWS CLI

Pour supprimer une définition de tâche de passerelle sans fil

L'`delete-wireless-gateway-task-definition`exemple suivant supprime la définition de tâche de passerelle sans fil que vous avez créée avec l'ID suivant.

```
aws iotwireless delete-wireless-gateway-task-definition \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteWirelessGatewayTaskDefinitionà la section Référence des AWS CLI commandes.

`delete-wireless-gateway-task`

L'exemple de code suivant montre comment utiliser`delete-wireless-gateway-task`.

AWS CLI

Pour supprimer une tâche de passerelle sans fil

L'`delete-wireless-gateway-task`exemple suivant supprime la tâche de passerelle sans fil dont l'ID est spécifié.

```
aws iotwireless delete-wireless-gateway-task \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteWirelessGatewayTaskà la section Référence des AWS CLI commandes.

`delete-wireless-gateway`

L'exemple de code suivant montre comment utiliser`delete-wireless-gateway`.

AWS CLI

Pour supprimer une passerelle sans fil

L'`delete-wireless-gateway`exemple suivant supprime une passerelle sans fil qui possède l'ID spécifié.

```
aws iotwireless delete-wireless-gateway \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DeleteWirelessGatewayà la section Référence des AWS CLI commandes.

`disassociate-aws-account-from-partner-account`

L'exemple de code suivant montre comment utiliser`disassociate-aws-account-from-partner-account`.

AWS CLI

Pour dissocier le compte partenaire du AWS compte

L'`disassociate-aws-account-from-partner-account`exemple suivant dissocie un compte partenaire de votre AWS compte actuellement associé.

```
aws iotwireless disassociate-aws-account-from-partner-account \
    --partner-account-id "12345678901234" \
    --partner-type "Sidewalk"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DisassociateAwsAccountFromPartnerAccountà la section Référence des AWS CLI commandes.

`disassociate-wireless-device-from-thing`

L'exemple de code suivant montre comment utiliser`disassociate-wireless-device-from-thing`.

AWS CLI

Pour dissocier l'objet du périphérique sans fil

L'`disassociate-wireless-device-from-thing`exemple suivant dissocie un périphérique sans fil de l'objet actuellement associé.

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DisassociateWirelessDeviceFromThingà la section Référence des AWS CLI commandes.

`disassociate-wireless-gateway-from-certificate`

L'exemple de code suivant montre comment utiliser`disassociate-wireless-gateway-from-certificate`.

AWS CLI

Pour dissocier le certificat de la passerelle sans fil

Ce qui suit `disassociate-wireless-gateway-from-certificate` dissocie une passerelle sans fil de son certificat actuellement associé.

```
aws iotwireless disassociate-wireless-gateway-from-certificate \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DisassociateWirelessGatewayFromCertificateà la section Référence des AWS CLI commandes.

`disassociate-wireless-gateway-from-thing`

L'exemple de code suivant montre comment utiliser`disassociate-wireless-gateway-from-thing`.

AWS CLI

Pour dissocier l'objet de la passerelle sans fil

L'`disassociate-wireless-gateway-from-thing`exemple suivant dissocie une passerelle sans fil de l'objet actuellement associé.

```
aws iotwireless disassociate-wireless-gateway-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter vos passerelles et appareils sans fil à AWS IoT Core for LoRa WAN dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous DisassociateWirelessGatewayFromThingà la section Référence des AWS CLI commandes.

`get-destination`

L'exemple de code suivant montre comment utiliser`get-destination`.

AWS CLI

Pour obtenir des informations sur une destination sans fil IoT

L'`get-destination`exemple suivant permet d'obtenir des informations sur la ressource de destination portant le nom `IoTWirelessDestination` que vous avez créé.

```
aws iotwireless get-destination \
    --name "IoTWirelessDestination"
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "IoTWirelessDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

Pour plus d'informations, consultez la section Ajouter des destinations à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetDestinationà la section Référence des AWS CLI commandes.

`get-device-profile`

L'exemple de code suivant montre comment utiliser`get-device-profile`.

AWS CLI

Pour obtenir des informations sur le profil d'un appareil

L'`get-device-profile`exemple suivant permet d'obtenir des informations sur le profil de l'appareil avec l'ID spécifié que vous avez créé.

```
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "LoRaWAN": {
    "MacVersion": "1.0.3",
    "MaxDutyCycle": 10,
    "Supports32BitFCnt": false,
    "RegParamsRevision": "RP002-1.0.1",
    "SupportsJoin": true,
    "RfRegion": "US915",
    "MaxEirp": 13,
    "SupportsClassB": false,
    "SupportsClassC": false
    }
}
```

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetDeviceProfileà la section Référence des AWS CLI commandes.

`get-partner-account`

L'exemple de code suivant montre comment utiliser`get-partner-account`.

AWS CLI

Pour obtenir les informations du compte partenaire

L'`get-partner-account`exemple suivant permet d'obtenir des informations sur votre compte Sidewalk qui possède l'identifiant suivant.

```
aws iotwireless get-partner-account \
    --partner-account-id "12345678901234" \
    --partner-type "Sidewalk"
```

Sortie :

```
{
    "Sidewalk": {
        "AmazonId": "12345678901234",
        "Fingerprint":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
    },
    "AccountLinked": false
}
```

Pour plus d'informations, consultez Amazon Sidewalk Integration for AWS IoT Core dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetPartnerAccountà la section Référence des AWS CLI commandes.

`get-service-endpoint`

L'exemple de code suivant montre comment utiliser`get-service-endpoint`.

AWS CLI

Pour obtenir le point de terminaison du service

L'`get-service-endpoint`exemple suivant obtient le point de terminaison spécifique au compte pour le protocole CUPS.

```
aws iotwireless get-service-endpoint
```

Sortie :

```
{
    "ServiceType": "CUPS",
    "ServiceEndpoint": "https://A1RMKZ37ACAGOT.cups.lorawan.us-
east-1.amazonaws.com:443",
```

```
    "ServerTrust": "-----BEGIN CERTIFICATE-----\n
    MIIESTCCAzGgAwIBAgITBn+UV4WH6Kx33rJTMlu8mYtWDTANBgkqhkiG9w0BAQsF\n
    ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6\n
    b24gUm9vdCBDQSAxMB4XDTE1MTAyMjAwMDAwMFoXDTI1MTAxOTAwMDAwMFowRjEL\n
    MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEVMBMGA1UECxMMU2VydmVyIENB\n
    IDFCMQ8wDQYDVQQDEwZBbWF6b24wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n
    AoIBAQDCThZn3c68asg3Wuw6MLAd5tES6BIoSMzoKcG5blPVo+sDORrMd4f2AbnZ\n
    cMzPa43j4wNxhplty6aUKk4T1qe9BOwKFjwK6zmxxLVYo7bHViXsPlJ6qOMpFge5\n
    blDP+18x+B26A0piiQOuPkfyDyeR4xQghfj66Yo19V+emU3nazfvpFA+ROz6WoVm\n
    B5x+F2pV8xeKNR7u6azDdU5YVX1TawprmxRC1+WsAYmz6qP+z8ArDITC2FMVy2fw\n
    0IjKOtEXc/VfmtTFch5+AfGYMGMqqvJ6LcXiAhqG5TI+Dr0RtM88k+8XUBCeQ8IG\n
    KuANaL7TiItKZYxK1MMuTJtV9IblAgMBAAGjggE7MIIBNzASBgNVHRMBAf8ECDAG\n
    AQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUWaRmBlKge5WSPKOUByeW\n
    dFv5PdAwHwYDVR0jBBgwFoAUhBjMhTTsvAyUlC4IWZzHshBOCggwewYIKwYBBQUH\n
    AQEEbzBtMC8GCCsGAQUFBzABhiNodHRwOi8vb2NzcC5yb290Y2ExLmFtYXpvbnRy\n
    dXN0LmNvbTA6BggrBgEFBQcwAoYuaHR0cDovL2NydC5yb290Y2ExLmFtYXpvbnRy\n
    dXN0LmNvbS9yb290Y2ExLmNlcjA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3Js\n
    LnJvb3RjYTEuYW1hem9udHJ1c3QuY29tL3Jvb3RjYTEuY3JsMBMGA1UdIAQMMAow\n
    CAYGZ4EMAQIBMA0GCSqGSIb3DQEBCwUAA4IBAQCFkr41u3nPo4FCHOTjY3NTOVI1\n
    59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss2lJwJFqMp2PphKg5625kXg8kP2CN5t\n
    6G7bMQcT8C8xDZNtYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n
    8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB48Oz+ctBWuZSkbvkNodPLamkB2g1\n
    upRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtKPISLMVCx7i4hncxHZS\n
    yLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n
    -----END CERTIFICATE-----\n"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetServiceEndpointà la section Référence des AWS CLI commandes.

`get-service-profile`

L'exemple de code suivant montre comment utiliser`get-service-profile`.

AWS CLI

Pour obtenir des informations sur un profil de service

L'`get-service-profile`exemple suivant permet d'obtenir des informations sur le profil de service avec l'ID spécifié que vous avez créé.

```
aws iotwireless get-service-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Sortie :

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:651419225604:ServiceProfile/538185bb-
d7e7-4b95-96a0-c51aa4a5b9a0",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "LoRaWAN": {
        "HrAllowed": false,
        "NwkGeoLoc": false,
        "DrMax": 15,
        "UlBucketSize": 4096,
        "PrAllowed": false,
        "ReportDevStatusBattery": false,
        "DrMin": 0,
        "DlRate": 60,
        "AddGwMetadata": false,
        "ReportDevStatusMargin": false,
        "MinGwDiversity": 1,
        "RaAllowed": false,
        "DlBucketSize": 4096,
        "DevStatusReqFreq": 24,
        "TargetPer": 5,
        "UlRate": 60
    }
}
```

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetServiceProfileà la section Référence des AWS CLI commandes.

`get-wireless-device-statistics`

L'exemple de code suivant montre comment utiliser`get-wireless-device-statistics`.

AWS CLI

Pour obtenir des informations sur le fonctionnement d'un appareil sans fil

L'`get-wireless-device-statistics`exemple suivant permet d'obtenir des informations de fonctionnement relatives à un périphérique sans fil.

```
aws iotwireless get-wireless-device-statistics \
    --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

Sortie :

```
{
    "WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessDeviceStatisticsà la section Référence des AWS CLI commandes.

`get-wireless-device`

L'exemple de code suivant montre comment utiliser`get-wireless-device`.

AWS CLI

Pour obtenir des informations sur le périphérique sans fil

L'`get-wireless-device`exemple suivant répertorie les widgets disponibles dans votre AWS compte.

```
aws iotwireless get-wireless-device \
    --identifier "1ffd32c8-8130-4194-96df-622f072a315f" \
    --identifier-type WirelessDeviceID
```

Sortie :

```
{
    "Name": "myLoRaWANDevice",
    "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/44b87eb4-9bce-423d-
b5fc-973f5ecc358b",
    "DestinationName": "IoTWirelessDestination",
```

```
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
    "ThingName": "44b87eb4-9bce-423d-b5fc-973f5ecc358b",
    "Type": "LoRaWAN",
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
    "Description": "My LoRaWAN wireless device"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessDeviceà la section Référence des AWS CLI commandes.

`get-wireless-gateway-certificate`

L'exemple de code suivant montre comment utiliser`get-wireless-gateway-certificate`.

AWS CLI

Pour obtenir l'ID d'un certificat associé à une passerelle sans fil

L'`get-wireless-gateway-certificate`exemple suivant obtient l'ID du certificat associé à une passerelle sans fil qui possède l'ID spécifié.

```
aws iotwireless get-wireless-gateway-certificate \
    --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

Sortie :

```
{
```

```
    "IotCertificateId":
 "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessGatewayCertificateà la section Référence des AWS CLI commandes.

`get-wireless-gateway-firmware-information`

L'exemple de code suivant montre comment utiliser`get-wireless-gateway-firmware-information`.

AWS CLI

Pour obtenir des informations sur le microprogramme d'une passerelle sans fil

L'`get-wireless-gateway-firmware-information`exemple suivant permet d'obtenir la version du microprogramme et d'autres informations sur une passerelle sans fil.

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Sortie :

```
{
    "LoRaWAN" :{
        "CurrentVersion" :{
            "PackageVersion" : "1.0.0",
            "Station" : "2.0.5",
            "Model" : "linux"
        }
    }
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessGatewayFirmwareInformationà la section Référence des AWS CLI commandes.

`get-wireless-gateway-statistics`

L'exemple de code suivant montre comment utiliser`get-wireless-gateway-statistics`.

AWS CLI

Pour obtenir des informations de fonctionnement concernant une passerelle sans fil

L'`get-wireless-gateway-statistics`exemple suivant permet d'obtenir des informations de fonctionnement concernant une passerelle sans fil.

```
aws iotwireless get-wireless-gateway-statistics \
    --wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Sortie :

```
{
    "WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessGatewayStatisticsà la section Référence des AWS CLI commandes.

`get-wireless-gateway-task-definition`

L'exemple de code suivant montre comment utiliser`get-wireless-gateway-task-definition`.

AWS CLI

Pour obtenir des informations sur la définition d'une tâche de passerelle sans fil

L'`get-wireless-gateway-task-definition`exemple suivant permet d'obtenir des informations sur la définition de la tâche sans fil avec l'ID spécifié.

```
aws iotwireless get-wireless-gateway-task-definition \
    --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

Sortie :

```
{
    "AutoCreateTasks": true,
    "Name": "TestAutoUpdate",
    "Update":{
        "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",
        "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",
        "LoRaWAN" :{
            "CurrentVersion" :{
                "PackageVersion" : "1.0.0",
                "Station" : "2.0.5",
                "Model" : "linux"
            },
            "UpdateVersion" :{
                "PackageVersion" : "1.0.1",
                "Station" : "2.0.5",
                "Model" : "minihub"
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessGatewayTaskDefinitionà la section Référence des AWS CLI commandes.

`get-wireless-gateway-task`

L'exemple de code suivant montre comment utiliser`get-wireless-gateway-task`.

AWS CLI

Pour obtenir des informations sur la tâche de passerelle sans fil

L'`get-wireless-gateway-task`exemple suivant permet d'obtenir des informations sur la tâche de passerelle sans fil avec l'ID spécifié.

```
aws iotwireless get-wireless-gateway-task \
    --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

Sortie :

```
{
    "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",
    "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
    "Status": "Success"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessGatewayTaskà la section Référence des AWS CLI commandes.

`get-wireless-gateway`

L'exemple de code suivant montre comment utiliser`get-wireless-gateway`.

AWS CLI

Pour obtenir des informations sur une passerelle sans fil

L'`get-wireless-gateway`exemple suivant permet d'obtenir des informations sur la passerelle sans fil`myFirstLoRaWANGateway`.

```
aws iotwireless get-wireless-gateway \
    --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --identifier-type WirelessGatewayId
```

Sortie :

```
{
    "Description": "My first LoRaWAN gateway",
    "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/a1b2c3d4-5678-90ab-
cdef-12ab345c67de",
    "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "a1b2c3d4567890ab"
    },
    "ThingName": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
```

```
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/6c44ab31-8b4d-407a-bed3-19b6c7cda551",
    "Name": "myFirstLoRaWANGateway"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous GetWirelessGatewayà la section Référence des AWS CLI commandes.

`list-destinations`

L'exemple de code suivant montre comment utiliser`list-destinations`.

AWS CLI

Pour répertorier les destinations sans fil

L'`list-destinations`exemple suivant répertorie les destinations disponibles enregistrées sur votre AWS compte.

```
aws iotwireless list-destinations
```

Sortie :

```
{
    "DestinationList": [
        {
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
            "Name": "IoTWirelessDestination",
            "Expression": "IoTWirelessRule",
            "Description": "Destination for messages processed using
 IoTWirelessRule",
            "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
        },
        {
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination2",
            "Name": "IoTWirelessDestination2",
```

```
            "Expression": "IoTWirelessRule2",
            "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
        }
    ]
}
```

Pour plus d'informations, consultez la section Ajouter des destinations à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListDestinationsà la section Référence des AWS CLI commandes.

`list-device-profiles`

L'exemple de code suivant montre comment utiliser`list-device-profiles`.

AWS CLI

Pour répertorier les profils d'appareils

L'`list-device-profiles`exemple suivant répertorie les profils d'appareils disponibles enregistrés sur votre AWS compte.

```
aws iotwireless list-device-profiles
```

Sortie :

```
{
    "DeviceProfileList": [
        {
            "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d"
        },
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/
a1b2c3d4-5678-90ab-cdef-12ab345c67de"
        }
    ]
}
```

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListDeviceProfilesà la section Référence des AWS CLI commandes.

`list-partner-accounts`

L'exemple de code suivant montre comment utiliser`list-partner-accounts`.

AWS CLI

Pour répertorier les comptes partenaires

L'`list-partner-accounts`exemple suivant répertorie les comptes partenaires disponibles associés à votre AWS compte.

```
aws iotwireless list-partner-accounts
```

Sortie :

```
{
    "Sidewalk": [
        {
            "AmazonId": "78965678771228",
            "Fingerprint":
 "bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
        },
        {
            "AmazonId": "89656787651228",
            "Fingerprint":
 "bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Sidewalk Integration for AWS IoT Core dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListPartnerAccountsà la section Référence des AWS CLI commandes.

`list-service-profiles`

L'exemple de code suivant montre comment utiliser`list-service-profiles`.

AWS CLI

Pour répertorier les profils de service

L'`list-service-profiles`exemple suivant répertorie les profils de service disponibles enregistrés sur votre AWS compte.

```
aws iotwireless list-service-profiles
```

Sortie :

```
{
    "ServiceProfileList": [
        {
            "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:ServiceProfile/538185bb-d7e7-4b95-96a0-c51aa4a5b9a0"
        },
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/
ea8bc823-5d13-472e-8d26-9550737d8100"
        }
    ]
}
```

Pour plus d'informations, consultez la section Ajouter des profils à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListServiceProfilesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises attribuées à la ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises attribuées à une ressource de destination sans fil.

```
aws iotwireless list-tags-for-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination"
```

Sortie :

```
{
    "Tags": [
        {
            "Value": "MyValue",
            "Key": "MyTag"
        }
    ]
}
```

Pour plus d'informations, consultez la section Décrire vos ressources AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-wireless-devices`

L'exemple de code suivant montre comment utiliser`list-wireless-devices`.

AWS CLI

Pour répertorier les appareils sans fil disponibles

L'`list-wireless-devices`exemple suivant répertorie les appareils sans fil disponibles enregistrés sur votre AWS compte.

```
aws iotwireless list-wireless-devices
```

Sortie :

```
{
```

```
    "WirelessDeviceList": [
        {
            "Name": "myLoRaWANDevice",
            "DestinationName": "IoTWirelessDestination",
            "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
            "Type": "LoRaWAN",
            "LoRaWAN": {
                "DevEui": "ac12efc654d23fc2"
            },
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
        }
    ]
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListWirelessDevicesà la section Référence des AWS CLI commandes.

`list-wireless-gateway-task-definitions`

L'exemple de code suivant montre comment utiliser`list-wireless-gateway-task-definitions`.

AWS CLI

Pour répertorier les définitions de tâches de passerelle sans fil

L'`list-wireless-gateway-task-definitions`exemple suivant répertorie les définitions de tâches de passerelle sans fil disponibles enregistrées sur votre AWS compte.

```
aws iotwireless list-wireless-gateway-task-definitions
```

Sortie :

```
{
    "TaskDefinitions": [
        {
            "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
            "LoRaWAN" :
```

```
                {
                "CurrentVersion" :{
                    "PackageVersion" : "1.0.0",
                    "Station" : "2.0.5",
                    "Model" : "linux"
                },
                "UpdateVersion" :{
                    "PackageVersion" : "1.0.1",
                    "Station" : "2.0.5",
                    "Model" : "minihub"
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListWirelessGatewayTaskDefinitionsà la section Référence des AWS CLI commandes.

`list-wireless-gateways`

L'exemple de code suivant montre comment utiliser`list-wireless-gateways`.

AWS CLI

Pour répertorier les passerelles sans fil

L'`list-wireless-gateways`exemple suivant répertorie les passerelles sans fil disponibles dans votre AWS compte.

```
aws iotwireless list-wireless-gateways
```

Sortie :

```
{
    "WirelessGatewayList": [
        {
            "Description": "My first LoRaWAN gateway",
            "LoRaWAN": {
```

```
                "RfRegion": "US915",
                "GatewayEui": "dac632ebc01d23e4"
            },
            "Id": "3039b406-5cc9-4307-925b-9948c63da25b",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/3039b406-5cc9-4307-925b-9948c63da25b",
            "Name": "myFirstLoRaWANGateway"
        },
        {
            "Description": "My second LoRaWAN gateway",
            "LoRaWAN": {
                "RfRegion": "US915",
                "GatewayEui": "cda123fffe92ecd2"
            },
            "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
            "Name": "mySecondLoRaWANGateway"
        }
    ]
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous ListWirelessGatewaysà la section Référence des AWS CLI commandes.

`send-data-to-wireless-device`

L'exemple de code suivant montre comment utiliser`send-data-to-wireless-device`.

AWS CLI

Pour envoyer des données à l'appareil sans fil

L'`send-data-to-wireless-device`exemple suivant envoie une trame de données d'application déchiffrée au périphérique sans fil.

```
aws iotwireless send-data-to-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
    --transmit-mode "1" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
```

```
    --wireless-metadata LoRaWAN={FPort=1}
```

Sortie :

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous SendDataToWirelessDeviceà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour spécifier une clé de balise et une valeur pour une ressource

L'`tag-resource`exemple suivant indique comment attribuer la clé `MyTag` et la valeur `IoTWirelessDestination` à la destination sans fil`MyValue`.

```
aws iotwireless tag-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/
IoTWirelessDestination" \
    --tags Key="MyTag",Value="MyValue"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Décrire vos ressources AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`test-wireless-device`

L'exemple de code suivant montre comment utiliser`test-wireless-device`.

AWS CLI

Pour tester le périphérique sans fil

L'`test-wireless-device`exemple suivant envoie des données de liaison montante `Hello` vers un appareil avec l'ID spécifié.

```
aws iotwireless test-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

Sortie :

```
{
    Result: "Test succeeded. one message is sent with payload: hello"
}
```

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous TestWirelessDeviceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une ou plusieurs balises d'une ressource

L'`untag-resource`exemple suivant supprime la balise `MyTag` et sa valeur de la destination sans fil`IoTWirelessDestination`.

```
aws iotwireless untag-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination" \
    --tag-keys "MyTag"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Décrire vos ressources AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-destination`

L'exemple de code suivant montre comment utiliser`update-destination`.

AWS CLI

Pour mettre à jour les propriétés d'une destination

L'`update-destination`exemple suivant met à jour la propriété de description d'une destination sans fil.

```
aws iotwireless update-destination \
    --name "IoTWirelessDestination" \
    --description "Destination for messages processed using IoTWirelessRule"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajouter des destinations à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateDestinationà la section Référence des AWS CLI commandes.

`update-partner-account`

L'exemple de code suivant montre comment utiliser`update-partner-account`.

AWS CLI

Pour mettre à jour les propriétés d'un compte partenaire

Ce qui suit `update-partner-account` met à jour le `AppServerPrivateKey` compte qui possède l'ID spécifié.

```
aws iotwireless update-partner-account \
    --partner-account-id "78965678771228" \
    --partner-type "Sidewalk" \
```

```
    --sidewalk
AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Amazon Sidewalk Integration for AWS IoT Core dans le guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdatePartnerAccountà la section Référence des AWS CLI commandes.

`update-wireless-device`

L'exemple de code suivant montre comment utiliser`update-wireless-device`.

AWS CLI

Pour mettre à jour les propriétés d'un appareil sans fil

L'`update-wireless-device`exemple suivant met à jour les propriétés d'un appareil sans fil enregistré sur votre AWS compte.

```
aws iotwireless update-wireless-device \
    --id "1ffd32c8-8130-4194-96df-622f072a315f" \
    --destination-name IoTWirelessDestination2 \
    --description "Using my first LoRaWAN device"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateWirelessDeviceà la section Référence des AWS CLI commandes.

`update-wireless-gateway`

L'exemple de code suivant montre comment utiliser`update-wireless-gateway`.

AWS CLI

Pour mettre à jour la passerelle sans fil

L'`update-wireless-gateway`exemple suivant met à jour la description de votre passerelle sans fil.

```
aws iotwireless update-wireless-gateway \
    --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \
    --description "Using my LoRaWAN gateway"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Connexion d'appareils et de passerelles à AWS IoT Core for LoRa WAN dans le Guide du développeur AWS IoT.

- Pour plus de détails sur l'API, reportez-vous UpdateWirelessGatewayà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon IVS AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon IVS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-get-channel`

L'exemple de code suivant montre comment utiliser`batch-get-channel`.

AWS CLI

Pour obtenir des informations de configuration de canal concernant plusieurs canaux

L'`batch-get-channel`exemple suivant répertorie les informations relatives aux canaux spécifiés.

```
aws ivs batch-get-channel \
    --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
        arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl
```

Sortie :

```
{
    "channels": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "authorized": false,
            "containerFormat": "TS",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "insecureIngest": false,
            "latencyMode": "LOW",
            "multitrackInputConfiguration": {
                "enabled": false,
                "maximumResolution": "FULL_HD",
                "policy": "ALLOW"
            },
            "name": "channel-1",
            "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/
api/video/v1/us-west-2.123456789012.channel-1.abcdEFGH.m3u8",
            "preset": "",
            "playbackRestrictionPolicyArn": "",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
            "srt": {
                "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
                "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
            },
            "tags": {},
            "type": "STANDARD"
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
            "authorized": false,
            "containerFormat": "FRAGMENTED_MP4",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
```

```
            "insecureIngest": false,
            "latencyMode": "LOW",
            "multitrackInputConfiguration": {
                "enabled": true,
                "maximumResolution": "FULL_HD",
                "policy": "ALLOW"
            },
            "name": "channel-2",
            "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/
api/video/v1/us-west-2.123456789012.channel-2.abcdEFGH.m3u8",
            "preset": "",
            "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"",
            "recordingConfigurationArn": "",
            "srt": {
                "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
                "passphrase":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
            },
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir BatchGetChannella section Référence des AWS CLI commandes.

`batch-get-stream-key`

L'exemple de code suivant montre comment utiliser`batch-get-stream-key`.

AWS CLI

Pour obtenir des informations sur plusieurs clés de flux

L'`batch-get-stream-key`exemple suivant permet d'obtenir des informations sur les clés de flux spécifiées.

```
aws ivs batch-get-stream-key \
```

```
    --arns arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh \
      arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop
```

Sortie :

```
{
    "streamKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",
            "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop",
            "value": "sk_us-west-2_abcdABCDefgh_567890ghijkl",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "tags": {}
        }
     ]
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir BatchGetStreamKeyla section Référence des AWS CLI commandes.

`batch-start-viewer-session-revocation`

L'exemple de code suivant montre comment utiliser`batch-start-viewer-session-revocation`.

AWS CLI

Pour révoquer les sessions de visionnage pour plusieurs paires canal-ARN et Viewer-ID

L'`batch-start-viewer-session-revocation`exemple suivant effectue la révocation de session sur plusieurs paires Channel-ARN et Viewer-ID simultanément. La demande peut se terminer normalement mais renvoyer des valeurs dans le champ d'erreurs si l'appelant n'est pas autorisé à révoquer la session spécifiée.

```
aws ivs batch-start-viewer-session-revocation \
    --viewer-sessions '[{"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1","viewerId":"abcdefg1","viewerSessionVersionsLessThanOrEqualTo":1234567890},
 \
     {"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2","viewerId":"abcdefg2","viewerSessionVersionsLessThanOrEqualTo":1234567890}]'
```

Sortie :

```
{
    "errors": [
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1",
            "viewerId": "abcdefg1",
            "code": "403",
            "message": "not authorized",
        },
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2",
            "viewerId": "abcdefg2",
            "code": "403",
            "message": "not authorized",
        }
    ]
}
```

Pour plus d'informations, reportez-vous à la section Configuration des canaux privés dans le Guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir BatchStartViewerSessionRevocationla section Référence des AWS CLI commandes.

`create-channel`

L'exemple de code suivant montre comment utiliser`create-channel`.

AWS CLI

Exemple 1 : pour créer une chaîne sans enregistrement

L'`create-channel`exemple suivant crée une nouvelle chaîne et une clé de diffusion associée pour démarrer le streaming.

```
aws ivs create-channel \
    --name 'test-channel' \
    --no-insecure-ingest
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "TS",
        "name": "test-channel",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
```

```
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

Exemple 2 : pour créer un canal avec l'enregistrement activé, en utilisant la RecordingConfiguration ressource spécifiée par son ARN

L'`create-channel`exemple suivant crée une nouvelle chaîne et une clé de diffusion associée pour démarrer le streaming, et configure l'enregistrement pour la chaîne :

```
aws ivs create-channel \
    --name test-channel-with-recording \
    --insecure-ingest \
    --recording-configuration-arn 'arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh'
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
```

```
        "preset": "",
        "authorized": false,
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur IVS Low-Latency.

Exemple 3 : pour créer une chaîne avec une politique de restriction de lecture spécifiée par son ARN

L'`create-channel`exemple suivant crée une nouvelle chaîne et une clé de diffusion associée pour démarrer le streaming, et définit une politique de restriction de lecture pour la chaîne :

```
aws ivs create-channel \
    --name test-channel-with-playback-restriction-policy\
    --insecure-ingest \
    --playback-restriction-policy-arn 'arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ'
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
```

```
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

Exemple 4 : pour créer une chaîne avec le multipiste activé

L'`create-channel`exemple suivant crée un nouveau canal et une clé de diffusion associée pour démarrer le streaming, et active le multipiste.

```
aws ivs create-channel \
    --name 'test-channel' \
    --no-insecure-ingest \
    --container-format 'FRAGMENTED_MP4' \
    --multitrack-input-configuration '{"enabled": true,"maximumResolution":
 "FULL_HD","policy": "ALLOW"}'
```

Sortie :

```
{
```

```
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "FRAGMENTED_MP4",
        "name": "test-channel",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": true,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir CreateChannella section Référence des AWS CLI commandes.

`create-playback-restriction-policy`

L'exemple de code suivant montre comment utiliser`create-playback-restriction-policy`.

AWS CLI

Pour créer une politique de restriction de lecture

L'`create-playback-restriction-policy`exemple suivant crée une nouvelle politique de restriction de lecture.

```
aws ivs create-playback-restriction-policy \
    --name "test-playback-restriction-policy" \
    --enable-strict-origin-enforcement \
    --tags "key1=value1, key2=value2" \
    --allowed-countries US MX \
    --allowed-origins https://www.website1.com https://www.website2.com
```

Sortie :

```
{
    "playbackRestrictionPolicy": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
        "allowedCountries": [
            "US",
            "MX"
        ],
        "allowedOrigins": [
            "https://www.website1.com",
            "https://www.website2.com"
        ],
        "enableStrictOriginEnforcement": true,
        "name": "test-playback-restriction-policy",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir CreatePlaybackRestrictionPolicyla section Référence des AWS CLI commandes.

`create-recording-configuration`

L'exemple de code suivant montre comment utiliser`create-recording-configuration`.

AWS CLI

Pour créer une RecordingConfiguration ressource

L'`create-recording-configuration`exemple suivant crée une RecordingConfiguration ressource pour activer l'enregistrement sur Amazon S3.

```
aws ivs create-recording-configuration \
    --name "test-recording-config" \
    --recording-reconnect-window-seconds 60 \
    --tags "key1=value1, key2=value2" \
    --rendition-configuration renditionSelection="CUSTOM",renditions="HD" \
    --thumbnail-configuration
 recordingMode="INTERVAL",targetIntervalSeconds=1,storage="LATEST",resolution="LOWEST_RESOLU
 \
    --destination-configuration s3={bucketName=demo-recording-bucket}
```

Sortie :

```
{
    "recordingConfiguration": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
        "name": "test-recording-config",
        "destinationConfiguration": {
            "s3": {
                "bucketName": "demo-recording-bucket"
            }
        },
        "state": "CREATING",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        },
        "thumbnailConfiguration": {
            "recordingMode": "INTERVAL",
            "targetIntervalSeconds": 1,
            "resolution": "LOWEST_RESOLUTION",
            "storage": [
```

```
                "LATEST"
            ]
        },
        "recordingReconnectWindowSeconds": 60,
        "renditionConfiguration": {
            "renditionSelection": "CUSTOM",
            "renditions": [
                "HD"
            ]
        }
    }
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir CreateRecordingConfigurationla section Référence des AWS CLI commandes.

`create-stream-key`

L'exemple de code suivant montre comment utiliser`create-stream-key`.

AWS CLI

Pour créer une clé de flux

L'`create-stream-key`exemple suivant crée une clé de flux pour un ARN (Amazon Resource Name) spécifié.

```
aws ivs create-stream-key \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Sortie :

```
{
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
```

```
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir CreateStreamKeyla section Référence des AWS CLI commandes.

`delete-channel`

L'exemple de code suivant montre comment utiliser`delete-channel`.

AWS CLI

Pour supprimer une chaîne et les clés de diffusion associées

L'`delete-channel`exemple suivant supprime le canal avec l'ARN (Amazon Resource Name) spécifié.

```
aws ivs delete-channel \
    --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir DeleteChannella section Référence des AWS CLI commandes.

`delete-playback-key-pair`

L'exemple de code suivant montre comment utiliser`delete-playback-key-pair`.

AWS CLI

Pour supprimer une paire de clés de lecture spécifiée

L'`delete-playback-key-pair`exemple suivant renvoie l'empreinte digitale de la paire de clés spécifiée.

```
aws ivs delete-playback-key-pair \
```

```
    --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, reportez-vous à la section Configuration des canaux privés dans le Guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir DeletePlaybackKeyPairla section Référence des AWS CLI commandes.

`delete-playback-restriction-policy`

L'exemple de code suivant montre comment utiliser`delete-playback-restriction-policy`.

AWS CLI

Pour supprimer une politique de restriction de lecture

L'`delete-playback-restriction-policy`exemple suivant supprime la politique de restriction de lecture avec l'ARN de politique spécifié (Amazon Resource Name).

```
aws ivs delete-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir DeletePlaybackRestrictionPolicyla section Référence des AWS CLI commandes.

`delete-recording-configuration`

L'exemple de code suivant montre comment utiliser`delete-recording-configuration`.

AWS CLI

Pour supprimer la RecordingConfiguration ressource spécifiée par son ARN

L'`delete-recording-configuration`exemple suivant supprime la RecordingConfiguration ressource avec l'ARN spécifié.

```
aws ivs delete-recording-configuration \
    --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir DeleteRecordingConfigurationla section Référence des AWS CLI commandes.

`delete-stream-key`

L'exemple de code suivant montre comment utiliser`delete-stream-key`.

AWS CLI

Pour supprimer une clé de diffusion

L'`delete-stream-key`exemple suivant supprime la clé de flux pour un ARN (Amazon Resource Name) spécifié, afin qu'il ne puisse plus être utilisé pour le streaming.

```
aws ivs delete-stream-key \
    --arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir DeleteStreamKeyla section Référence des AWS CLI commandes.

`get-channel`

L'exemple de code suivant montre comment utiliser`get-channel`.

AWS CLI

Pour obtenir les informations de configuration d'une chaîne

L'`get-channel`exemple suivant obtient la configuration du canal pour un ARN de canal spécifié (Amazon Resource Name).

```
aws ivs get-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh'
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "TS",
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "name": "channel-1",
        "playbackRestrictionPolicyArn": "",
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "tags": {}
        "type": "STANDARD",
    }
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir GetChannella section Référence des AWS CLI commandes.

`get-playback-key-pair`

L'exemple de code suivant montre comment utiliser`get-playback-key-pair`.

AWS CLI

Pour obtenir une paire de clés de lecture spécifiée

L'`get-playback-key-pair`exemple suivant renvoie l'empreinte digitale de la paire de clés spécifiée.

```
aws ivs get-playback-key-pair \
    --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

Sortie :

```
{
    "keyPair": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
        "name": "my-playback-key",
        "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
        "tags": {}
    }
}
```

Pour plus d'informations, reportez-vous à la section Configuration des canaux privés dans le Guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir GetPlaybackKeyPairla section Référence des AWS CLI commandes.

`get-playback-restriction-policy`

L'exemple de code suivant montre comment utiliser`get-playback-restriction-policy`.

AWS CLI

Pour obtenir les informations de configuration d'une politique de restriction de lecture

L'`get-playback-restriction-policy`exemple suivant obtient la configuration de la politique de restriction de lecture avec l'ARN de politique spécifié (Amazon Resource Name).

```
aws ivs get-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ"
```

Sortie :

```
{
    "playbackRestrictionPolicy": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
        "allowedCountries": [
            "US",
            "MX"
        ],
        "allowedOrigins": [
            "https://www.website1.com",
            "https://www.website2.com"
        ],
        "enableStrictOriginEnforcement": true,
        "name": "test-playback-restriction-policy",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir GetPlaybackRestrictionPolicyla section Référence des AWS CLI commandes.

`get-recording-configuration`

L'exemple de code suivant montre comment utiliser`get-recording-configuration`.

AWS CLI

Pour obtenir des informations sur une RecordingConfiguration ressource

L'`get-recording-configuration`exemple suivant obtient des informations sur la RecordingConfiguration ressource pour l'ARN spécifié.

```
aws ivs get-recording-configuration \
    --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

Sortie :

```
{
    "recordingConfiguration": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
        "destinationConfiguration": {
            "s3": {
                "bucketName": "demo-recording-bucket"
            }
        },
        "name": "test-recording-config",
        "recordingReconnectWindowSeconds": 60,
        "state": "ACTIVE",
        "tags": {
            "key1" : "value1",
            "key2" : "value2"
        },
        "thumbnailConfiguration": {
            "recordingMode": "INTERVAL",
            "targetIntervalSeconds": 1,
            "resolution": "LOWEST_RESOLUTION",
            "storage": [
                "LATEST"
            ]
        },
        "renditionConfiguration": {
            "renditionSelection": "CUSTOM",
            "renditions": [
                "HD"
            ]
        }
    }
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir GetRecordingConfigurationla section Référence des AWS CLI commandes.

`get-stream-key`

L'exemple de code suivant montre comment utiliser`get-stream-key`.

AWS CLI

Pour obtenir des informations sur un stream

L'`get-stream-key`exemple suivant permet d'obtenir des informations sur la clé de flux spécifiée.

```
aws ivs get-stream-key \
    --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-
west-2
```

Sortie :

```
{
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir GetStreamKeyla section Référence des AWS CLI commandes.

`get-stream-session`

L'exemple de code suivant montre comment utiliser`get-stream-session`.

AWS CLI

Pour obtenir les métadonnées d'un flux spécifique

L'`get-stream-session`exemple suivant obtient la configuration des métadonnées pour l'ARN du canal spécifié (Amazon Resource Name) et le flux spécifié ; si ce n'`streamId`est pas le cas, le flux le plus récent du canal est sélectionné.

```
aws ivs get-stream-session \
    --channel-arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --stream-id 'mystream'
```

Sortie :

```
{
    "streamSession": {
        "streamId": "mystream1",
        "startTime": "2023-06-26T19:09:28+00:00",
        "channel": {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "name": "mychannel",
            "latencyMode": "LOW",
            "type": "STANDARD",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABcdef34ghIJ",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "playbackUrl": "url-string",
            "authorized": false,
            "insecureIngest": false,
            "preset": ""
        },
        "ingestConfiguration": {
            "audio": {
                "channels": 2,
                "codec": "mp4a.40.2",
                "sampleRate": 8000,
                "targetBitrate": 46875,
                "track": "Track0"
            },
            "video": {
                "avcProfile": "Baseline",
                "avcLevel": "4.2",
                "codec": "avc1.42C02A",
```

```
                "encoder": "Lavf58.45.100",
                "level": "4.2",
                "profile": "Baseline",
                "targetBitrate": 8789062,
                "targetFramerate": 60,
                "track": "Track0",
                "videoHeight": 1080,
                "videoWidth": 1920
            }
        },
        "ingestConfigurations": {
            "audioConfigurations": [
                {
                    "channels": 2,
                    "codec": "mp4a.40.2",
                    "sampleRate": 8000,
                    "targetBitrate": 46875,
                    "track": "Track0"
                }
            ],
            "videoConfigurations": [
                {
                    "codec": "avc1.42C02A",
                    "encoder": "Lavf58.45.100",
                    "level": "4.2",
                    "profile": "Baseline",
                    "targetBitrate": 8789062,
                    "targetFramerate": 60,
                    "track": "Track0",
                    "videoHeight": 1080,
                    "videoWidth": 1920
                }
            ]
        },
        "recordingConfiguration": {
            "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
            "name": "test-recording-config",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-recording-bucket"
                }
            },
            "state": "ACTIVE",
```

```
            "tags": {
                "key1": "value1",
                "key2": "value2"
            },
            "thumbnailConfiguration": {
                "recordingMode": "INTERVAL",
                "targetIntervalSeconds": 1,
                "resolution": "LOWEST_RESOLUTION",
                "storage": [
                    "LATEST"
                ]
            },
            "recordingReconnectWindowSeconds": 60,
            "renditionConfiguration": {
                "renditionSelection": "CUSTOM",
                "renditions": [
                    "HD"
                ]
            }
        },
        "truncatedEvents": [
            {
                "code": "StreamTakeoverInvalidPriority",
                "name": "Stream Takeover Failure",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:48+00:00"
            },
            {
                "name": "Stream Takeover",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:47+00:00"
            },
            {
                "name": "Recording Start",
                "type": "IVS Recording State Change",
                "eventTime": "2023-06-26T19:09:35+00:00"
            },
            {
                "name": "Stream Start",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:34+00:00"
            },
            {
                "name": "Session Created",
```

```
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:28+00:00"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir GetStreamSessionla section Référence des AWS CLI commandes.

`get-stream`

L'exemple de code suivant montre comment utiliser`get-stream`.

AWS CLI

Pour obtenir des informations sur un stream

L'`get-stream`exemple suivant permet d'obtenir des informations sur le flux pour le canal spécifié.

```
aws ivs get-stream \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Sortie :

```
{
    "stream": {
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "startTime": "2020-05-05T21:55:38Z",
        "state": "LIVE",
        "health": "HEALTHY",
        "streamId": "st-ABCDEfghij01234KLMN5678",
        "viewerCount": 1
    }
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir GetStreamla section Référence des AWS CLI commandes.

`import-playback-key-pair`

L'exemple de code suivant montre comment utiliser`import-playback-key-pair`.

AWS CLI

Pour importer la partie publique d'une nouvelle paire de clés

L'`import-playback-key-pair`exemple suivant importe la clé publique spécifiée (spécifiée sous forme de chaîne au format PEM) et renvoie l'ARN et l'empreinte de la nouvelle paire de clés.

```
aws ivs import-playback-key-pair \
    --name "my-playback-key" \
    --public-key-material "G1lbnQxOTA3BgNVBAMMMFdoeSBhcmUgeW91IGRl..."
```

Sortie :

```
{
    "keyPair": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
        "name": "my-playback-key",
        "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
        "tags": {}
    }
}
```

Pour plus d'informations, reportez-vous à la section Configuration des canaux privés dans le Guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir ImportPlaybackKeyPairla section Référence des AWS CLI commandes.

`list-channels`

L'exemple de code suivant montre comment utiliser`list-channels`.

AWS CLI

Exemple 1 : Pour obtenir des informations récapitulatives sur toutes les chaînes

L'`list-channels`exemple suivant répertorie toutes les chaînes associées à votre AWS compte.

```
aws ivs list-channels
```

Sortie :

```
{
    "channels": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "name": "channel-1",
            "latencyMode": "LOW",
            "authorized": false,
            "insecureIngest": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
            "tags": {},
            "type": "STANDARD"
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
            "name": "channel-2",
            "latencyMode": "LOW",
            "authorized": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
            "recordingConfigurationArn": "",
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

Exemple 2 : pour obtenir des informations récapitulatives sur tous les canaux, filtrées selon l' RecordingConfiguration ARN spécifié

L'`list-channels`exemple suivant répertorie tous les canaux associés à l' RecordingConfiguration ARN spécifié pour votre AWS compte.

```
aws ivs list-channels \
    --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"
```

Sortie :

```
{
    "channels": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "name": "channel-1",
            "latencyMode": "LOW",
            "authorized": false,
            "insecureIngest": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur IVS Low-Latency.

Exemple 3 : pour obtenir des informations récapitulatives sur tous les canaux, filtrées selon l' PlaybackRestrictionPolicy ARN spécifié

L'`list-channels`exemple suivant répertorie tous les canaux associés à l' PlaybackRestrictionPolicy ARN spécifié pour votre AWS compte.

```
aws ivs list-channels \
    --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

Sortie :

```
{
    "channels": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
            "name": "channel-2",
            "latencyMode": "LOW",
            "authorized": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
            "recordingConfigurationArn": "",
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir ListChannelsla section Référence des AWS CLI commandes.

`list-playback-key-pairs`

L'exemple de code suivant montre comment utiliser`list-playback-key-pairs`.

AWS CLI

Pour obtenir des informations récapitulatives sur toutes les paires de touches de lecture

L'`list-playback-key-pairs`exemple suivant renvoie des informations sur toutes les paires de clés.

```
aws ivs list-playback-key-pairs
```

Sortie :

```
{
    "keyPairs": [
```

```
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
            "name": "test-key-0",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/ijkl5678mnop",
            "name": "test-key-1",
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, reportez-vous à la section Configuration des canaux privés dans le Guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir ListPlaybackKeyPairsla section Référence des AWS CLI commandes.

`list-playback-restriction-policies`

L'exemple de code suivant montre comment utiliser`list-playback-restriction-policies`.

AWS CLI

Pour obtenir des informations récapitulatives sur toutes les politiques de restriction de lecture

L'`list-playback-restriction-policies`exemple suivant répertorie toutes les politiques de restriction de lecture applicables à votre AWS compte.

```
aws ivs list-playback-restriction-policies
```

Sortie :

```
{
    "playbackRestrictionPolicies": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
            "allowedCountries": [
                "US",
```

```
                "MX"
            ],
            "allowedOrigins": [
                "https://www.website1.com",
                "https://www.website2.com"
            ],
            "enableStrictOriginEnforcement": true,
            "name": "test-playback-restriction-policy",
            "tags": {
                "key1": "value1",
                "key2": "value2"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir ListPlaybackRestrictionPolicieslа section Référence des AWS CLI commandes.

`list-recording-configurations`

L'exemple de code suivant montre comment utiliser`list-recording-configurations`.

AWS CLI

Pour répertorier toutes les RecordingConfiguration ressources créées dans ce compte

L'`list-recording-configurations`exemple suivant permet d'obtenir des informations sur toutes les RecordingConfiguration ressources de votre compte.

```
aws ivs list-recording-configurations
```

Sortie :

```
{
    "recordingConfigurations": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
```

```
            "name": "test-recording-config-1",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-recording-bucket-1"
                }
            },
            "state": "ACTIVE",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
CD12abcdGHIJ",
            "name": "test-recording-config-2",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-recording-bucket-2"
                }
            },
            "state": "ACTIVE",
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir ListRecordingConfigurationsla section Référence des AWS CLI commandes.

`list-stream-keys`

L'exemple de code suivant montre comment utiliser`list-stream-keys`.

AWS CLI

Pour obtenir une liste des clés de diffusion

L'`list-stream-keys`exemple suivant répertorie toutes les clés de flux pour un ARN (Amazon Resource Name) spécifié.

```
aws ivs list-stream-keys \
```

```
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Sortie :

```
{
    "streamKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "tags": {}
        }
    ]
}
```

FFor Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir ListStreamKeysla section Référence des AWS CLI commandes.

`list-stream-sessions`

L'exemple de code suivant montre comment utiliser`list-stream-sessions`.

AWS CLI

Pour obtenir un résumé des diffusions actuelles et précédentes pour une chaîne spécifique dans la AWS région actuelle

L'`list-stream-sessions`exemple suivant fournit des informations récapitulatives pour les flux pour un canal ARN (Amazon Resource Name) spécifié.

```
aws ivs list-stream-sessions \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --max-results 25 \
    --next-token ""
```

Sortie :

```
{
```

```
    "nextToken": "set-2",
    "streamSessions": [
        {
            "startTime": 1641578182,
            "endTime": 1641579982,
            "hasErrorEvent": false,
            "streamId": "mystream"
        }
        ...
    ]
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir ListStreamSessionsla section Référence des AWS CLI commandes.

`list-streams`

L'exemple de code suivant montre comment utiliser`list-streams`.

AWS CLI

Pour obtenir une liste des diffusions en direct et leur état

L'`list-streams`exemple suivant répertorie toutes les diffusions en direct de votre AWS compte.

```
aws ivs list-streams
```

Sortie :

```
{
   "streams": [
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "state": "LIVE",
            "health": "HEALTHY",
            "streamId": "st-ABCDEfghij01234KLMN5678",
            "viewerCount": 1
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir ListStreamsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier tous les tags d'une AWS ressource (par exemple : channel, stream key)

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises pour un ARN de ressource spécifié (Amazon Resource Name).

```
aws ivs list-tags-for-resource \
    --resource-arn arn:aws:ivs:us-west-2:12345689012:channel/abcdABCDefgh
```

Sortie :

```
{
    "tags":
    {
        "key1": "value1",
        "key2": "value2"
    }
}
```

Pour plus d'informations, consultez la section Balisage dans le manuel Amazon Interactive Video Service API Reference.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`put-metadata`

L'exemple de code suivant montre comment utiliser`put-metadata`.

AWS CLI

Pour insérer des métadonnées dans le flux actif pour un canal spécifique

L'`put-metadata`exemple suivant insère les métadonnées données dans le flux pour le canal spécifié.

```
aws ivs put-metadata \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --metadata '{"my": "metadata"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir PutMetadatala section Référence des AWS CLI commandes.

`start-viewer-session-revocation`

L'exemple de code suivant montre comment utiliser`start-viewer-session-revocation`.

AWS CLI

Pour révoquer une session de visionnage pour une paire ARN-canal et Viewer-ID donnée

L'`start-viewer-session-revocation`exemple suivant lance le processus de révocation de la session de visionnage associée à un ARN de canal et à un ID de lecteur spécifiés, jusqu'au numéro de version de session spécifié inclus. Si la version n'est pas fournie, la valeur par défaut est 0.

```
aws ivs batch-start-viewer-session-revocation \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --viewer-id abcdefg \
    --viewer-session-versions-less-than-or-equal-to 1234567890
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, reportez-vous à la section Configuration des canaux privés dans le Guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir StartViewerSessionRevocationla section Référence des AWS CLI commandes.

`stop-stream`

L'exemple de code suivant montre comment utiliser`stop-stream`.

AWS CLI

Pour arrêter un flux spécifique

L'`stop-stream`exemple suivant arrête le flux sur le canal spécifié.

```
aws ivs stop-stream \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir StopStreamla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter ou mettre à jour des balises pour une AWS ressource (par exemple : canal, clé de flux)

L'`tag-resource`exemple suivant ajoute ou met à jour des balises pour un ARN de ressource spécifié (Amazon Resource Name).

```
aws ivs tag-resource \
    --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le manuel Amazon Interactive Video Service API Reference.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les balises d'une AWS ressource (par exemple : canal, clé de flux)

L'`untag-resource`exemple suivant supprime les balises spécifiées pour un ARN de ressource spécifié (Amazon Resource Name).

```
aws ivs untag-resource \
    --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --tag-keys "tagkey1, tagkey2"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le manuel Amazon Interactive Video Service API Reference.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-channel`

L'exemple de code suivant montre comment utiliser`update-channel`.

AWS CLI

Exemple 1 : pour mettre à jour les informations de configuration d'un canal

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin de modifier le nom du canal. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --name 'channel-1' \
    --insecure-ingest
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "channel-1",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
}
```


Pour plus d'informations, consultez la section Créer un canal dans le guide de l'utilisateur IVS Low-Latency.

Exemple 2 : pour mettre à jour la configuration d'une chaîne afin d'activer l'enregistrement

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin de permettre l'enregistrement. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --no-insecure-ingest \
    --recording-configuration-arn 'arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh'
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur IVS Low-Latency.

Exemple 3 : pour mettre à jour la configuration d'une chaîne afin de désactiver l'enregistrement

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin de désactiver l'enregistrement. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
```

```
    --recording-configuration-arn ''
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Enregistrer sur Amazon S3 dans le guide de l'utilisateur IVS Low-Latency.

Exemple 4 : pour mettre à jour la configuration d'une chaîne afin d'activer la restriction de lecture

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin d'appliquer une politique de restriction de lecture. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
```

```
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --no-insecure-ingest \
    --playback-restriction-policy-arn 'arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ'
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

Exemple 5 : Pour mettre à jour la configuration d'une chaîne afin de désactiver la restriction de lecture

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin de désactiver la restriction de lecture. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --playback-restriction-policy-arn ''
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDeFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

Exemple 6 : Pour mettre à jour la configuration d'un canal afin d'activer le multipiste

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin d'activer le multipiste. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --container-format 'FRAGMENTED_MP4' \
    --multitrack-input-configuration '{"enabled": true,"maximumResolution":
 "FULL_HD","policy": "ALLOW"}'
```

Sortie :

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "FRAGMENTED_MP4",
        "name": "test-channel-with-multitrack",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": true,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

Exemple 7 : Pour mettre à jour la configuration d'une chaîne afin de désactiver la restriction de lecture

L'`update-channel`exemple suivant met à jour la configuration du canal pour un ARN de canal spécifié afin de désactiver le multipiste. Cela n'affecte pas le flux continu de cette chaîne ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --container-format 'TS' \
    --multitrack-input-configuration '{"enabled": false}'
```

Sortie :

```
{
    "channel": {
        ""arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-multitrack",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
```

```
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir UpdateChannella section Référence des AWS CLI commandes.

`update-playback-restriction-policy`

L'exemple de code suivant montre comment utiliser`update-playback-restriction-policy`.

AWS CLI

Pour mettre à jour une politique de restriction de lecture

L'`update-playback-restriction-policy`exemple suivant met à jour la politique de restriction de lecture avec l'ARN de politique spécifié afin de désactiver l'application stricte de l'origine. Cela n'affecte pas le flux en cours de la chaîne associée ; vous devez arrêter et redémarrer le flux pour que les modifications prennent effet.

```
aws ivs update-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ" \
    --no-enable-strict-origin-enforcement
```

Sortie :

```
{
    "playbackRestrictionPolicy": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
        "allowedCountries": [
            "US",
            "MX"
        ],
        "allowedOrigins": [
            "https://www.website1.com",
            "https://www.website2.com"
```

```
        ],
        "enableStrictOriginEnforcement": false,
        "name": "test-playback-restriction-policy",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

Pour plus d'informations, consultez la section Contenu et spectateurs indésirables dans le guide de l'utilisateur IVS Low-Latency.

- Pour plus de détails sur l'API, voir UpdatePlaybackRestrictionPolicyla section Référence des AWS CLI commandes.

Exemples d'Amazon IVS Chat utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon IVS Chat.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-chat-token`

L'exemple de code suivant montre comment utiliser`create-chat-token`.

AWS CLI

Pour créer un jeton de chat

L'`create-chat-token`exemple suivant crée un jeton de chat crypté qui est utilisé pour établir une WebSocket connexion individuelle à un salon. Le jeton est valide pendant une minute, et une connexion (session) établie avec le jeton est valide pendant la durée spécifiée.

```
aws ivschat create-chat-token \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \
    --userId" "11231234" \
    --capabilities "SEND_MESSAGE", \
    --sessionDurationInMinutes" 30
```

Sortie :

```
{
    "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",
    "sessionExpirationTime": "2022-03-16T04:44:09+00:00"
    "state": "CREATING",
    "tokenExpirationTime": "2022-03-16T03:45:09+00:00"
}
```

Pour plus d'informations, consultez Étape 3 : Authentifier et autoriser les clients de chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir CreateChatTokenla section Référence des AWS CLI commandes.

`create-logging-configuration`

L'exemple de code suivant montre comment utiliser`create-logging-configuration`.

AWS CLI

Pour créer une LoggingConfiguration ressource de chat

L'`create-logging-configuration`exemple suivant crée une LoggingConfiguration ressource qui permet aux clients de stocker et d'enregistrer les messages envoyés.

```
aws ivschat create-logging-configuration \
    --destination-configuration s3={bucketName=demo-logging-bucket} \
    --name "test-logging-config" \
    --tags "key1=value1, key2=value2"
```

Sortie :

```
{
    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-logging-bucket"
        }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir CreateLoggingConfigurationla section Référence des AWS CLI commandes.

`create-room`

L'exemple de code suivant montre comment utiliser`create-room`.

AWS CLI

Pour créer une pièce

L'`create-room`exemple suivant crée une nouvelle pièce.

```
aws ivschat create-room \
    --name "test-room-1" \
    --logging-configuration-identifiers "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
    --maximum-message-length 256 \
    --maximum-message-rate-per-second 5
```

Sortie :

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
    "id": "g1H2I3j4k5L6",
    "createTime": "2022-03-16T04:44:09+00:00",
    "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
    "maximumMessageLength": 256,
    "maximumMessageRatePerSecond": 5,
    "name": "test-room-1",
    "tags": {}
    "updateTime": "2022-03-16T07:22:09+00:00"
}
```

Pour plus d'informations, consultez l'étape 2 : créer un salon de discussion dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir CreateRoomla section Référence des AWS CLI commandes.

`delete-logging-configuration`

L'exemple de code suivant montre comment utiliser`delete-logging-configuration`.

AWS CLI

Pour supprimer une LoggingConfiguration ressource de chat

L'`delete-logging-configuration`exemple suivant supprime la LoggingConfiguration ressource pour l'ARN spécifié.

```
aws ivschat delete-logging-configuration \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir DeleteLoggingConfigurationla section Référence des AWS CLI commandes.

`delete-message`

L'exemple de code suivant montre comment utiliser`delete-message`.

AWS CLI

Pour supprimer des messages d'une pièce spécifiée

L'`delete-message`exemple suivant envoie un événement à la salle spécifiée, qui demande aux clients de supprimer le message spécifié, c'est-à-dire de le supprimer de la vue et de le supprimer de l'historique des discussions du client.

```
aws ivschat delete-message \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --id "ABC123def456" \
    --reason "Message contains profanity"
```

Sortie :

```
{
    "id": "12345689012"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir DeleteMessagela section Référence des AWS CLI commandes.

`delete-room`

L'exemple de code suivant montre comment utiliser`delete-room`.

AWS CLI

Pour supprimer une chambre

L'`delete-room`exemple suivant supprime la pièce spécifiée. Les clients connectés sont déconnectés. En cas de succès, il renvoie HTTP 204 avec un corps de réponse vide.

```
aws ivschat delete-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir DeleteRoomla section Référence des AWS CLI commandes.

`disconnect-user`

L'exemple de code suivant montre comment utiliser`disconnect-user`.

AWS CLI

Pour déconnecter un utilisateur d'une salle

L'`disconnect-user`exemple suivant déconnecte toutes les connexions de l'utilisateur spécifié à partir de la pièce spécifiée. En cas de succès, il renvoie HTTP 200 avec un corps de réponse vide.

```
aws ivschat disconnect-user \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --userId "ABC123def456" \
    --reason "Violated terms of service"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir DisconnectUserla section Référence des AWS CLI commandes.

`get-logging-configuration`

L'exemple de code suivant montre comment utiliser`get-logging-configuration`.

AWS CLI

Pour obtenir des informations sur une LoggingConfiguration ressource

L'`get-logging-configuration`exemple suivant permet d'obtenir des informations sur la LoggingConfiguration ressource pour l'ARN spécifié.

```
aws ivschat get-logging-configuration \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ"
```

Sortie :

```
{
    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-logging-bucket"
        }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir GetLoggingConfigurationla section Référence des AWS CLI commandes.

`get-room`

L'exemple de code suivant montre comment utiliser`get-room`.

AWS CLI

Pour obtenir la chambre spécifiée

L'`get-room`exemple suivant permet d'obtenir des informations sur la pièce spécifiée.

```
aws ivschat get-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Sortie :

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
    "createTime": "2022-03-16T04:44:09+00:00",
    "id": "g1H2I3j4k5L6",
    "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
    "maximumMessageLength": 256,
    "maximumMessageRatePerSecond": 5,
    "name": "test-room-1",
    "tags": {},
    "updateTime": "2022-03-16T07:22:09+00:00"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir GetRoomla section Référence des AWS CLI commandes.

`list-logging-configurations`

L'exemple de code suivant montre comment utiliser`list-logging-configurations`.

AWS CLI

Pour obtenir des informations récapitulatives sur toutes les configurations de journalisation pour l'utilisateur dans la AWS région où la demande d'API est traitée

L'`list-logging-configurations`exemple suivant répertorie les informations relatives à toutes les LoggingConfiguration ressources destinées à l'utilisateur dans la AWS région où la demande d'API est traitée.

```
aws ivschat list-logging-configurations \
    --max-results 2 \
    --next-token ""
```

Sortie :

```
{
    "nextToken": "set-2",
    "loggingConfigurations": [
```

```
        {
            "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
            "createTime": "2022-09-14T17:48:00.653000+00:00",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-logging-bucket"
                }
            },
            "id": "ABcdef34ghIJ",
            "name": "test-logging-config",
            "state": "ACTIVE",
            "tags": { "key1" : "value1", "key2" : "value2" },
            "updateTime": "2022-09-14T17:48:01.104000+00:00"
        }
        ...
    ]
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir ListLoggingConfigurationsla section Référence des AWS CLI commandes.

`list-rooms`

L'exemple de code suivant montre comment utiliser`list-rooms`.

AWS CLI

Pour obtenir des informations récapitulatives sur toutes vos chambres dans la région actuelle

L'`list-rooms`exemple suivant permet d'obtenir des informations récapitulatives sur toutes les pièces de la AWS région où la demande est traitée. Les résultats sont triés par ordre décroissant de UpdateTime.

```
aws ivschat list-rooms \
    --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
    --max-results 10 \
    --next-token ""
```

Sortie :

```
{
    "nextToken": "page3",
    "rooms": [
        {
            "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
            "createTime": "2022-03-16T04:44:09+00:00",
            "id": "g1H2I3j4k5L6",
            "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
            "name": "test-room-1",
            "tags": {},
            "updateTime": "2022-03-16T07:22:09+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir ListRoomsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier tous les tags d'une AWS ressource (par exemple : Room)

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées à un ARN de ressource spécifié (Amazon Resource Name).

```
aws ivschat list-tags-for-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6
```

Sortie :

```
{
    "tags":
    {
```

```
        "key1": "value1",
        "key2": "value2"
    }
}
```

Pour plus d'informations, consultez la section Balisage dans le manuel Amazon Interactive Video Service API Reference.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`send-event`

L'exemple de code suivant montre comment utiliser`send-event`.

AWS CLI

Pour envoyer un événement dans une salle

L'`send-event`exemple suivant envoie l'événement donné à la salle spécifiée.

```
aws ivschat send-event \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --eventName "SystemMessage" \
    --attributes \
        "msgType"="user-notification", \
        "msgText"="This chat room will close in 15 minutes."
```

Sortie :

```
{
    "id": "12345689012"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir SendEventla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter ou mettre à jour des balises pour une AWS ressource (par exemple : Room)

L'`tag-resource`exemple suivant ajoute ou met à jour des balises pour un ARN de ressource spécifié (Amazon Resource Name). En cas de succès, il renvoie HTTP 200 avec un corps de réponse vide.

```
aws ivschat tag-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \
    --tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le manuel Amazon Interactive Video Service API Reference.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les balises d'une AWS ressource (par exemple : Room)

L'`untag-resource`exemple suivant supprime les balises spécifiées pour un ARN de ressource spécifié (Amazon Resource Name). En cas de succès, il renvoie HTTP 200 avec un corps de réponse vide.

```
aws ivschat untag-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \
    --tag-keys "tagkey1, tagkey2"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Balisage dans le manuel Amazon Interactive Video Service API Reference.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-logging-configuration`

L'exemple de code suivant montre comment utiliser`update-logging-configuration`.

AWS CLI

Pour mettre à jour la configuration de journalisation d'une salle

L'`update-logging-configuration`exemple suivant met à jour une LoggingConfiguration ressource avec les données données.

```
aws ivschat update-logging-configuration \
    --destination-configuration s3={bucketName=demo-logging-bucket} \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ" \
    --name "test-logging-config"
```

Sortie :

```
{
    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-logging-bucket"
        }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir UpdateLoggingConfigurationla section Référence des AWS CLI commandes.

`update-room`

L'exemple de code suivant montre comment utiliser`update-room`.

AWS CLI

Pour mettre à jour la configuration d'une pièce

L'`update-room`exemple suivant met à jour la configuration de la pièce spécifiée avec les données fournies.

```
aws ivschat update-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --logging-configuration-identifiers "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
    --name "chat-room-a" \
    --maximum-message-length 256 \
    --maximum-message-rate-per-second 5
```

Sortie :

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
    "createTime": "2022-03-16T04:44:09+00:00",
    "id": "g1H2I3j4k5L6",
    "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
    "maximumMessageLength": 256,
    "maximumMessageRatePerSecond": 5,
    "name": "chat-room-a",
    "tags": {},
    "updateTime": "2022-03-16T07:22:09+00:00"
}
```

Pour plus d'informations, consultez Getting Started with Amazon IVS Chat dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, voir UpdateRoomla section Référence des AWS CLI commandes.

Exemples de diffusion en temps réel d'Amazon IVS à l'aide d'Amazon IVS AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants en utilisant le streaming en temps réel AWS Command Line Interface avec Amazon IVS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-encoder-configuration`

L'exemple de code suivant montre comment utiliser`create-encoder-configuration`.

AWS CLI

Pour créer une configuration d'encodeur de composition

L'`create-encoder-configuration`exemple suivant crée une configuration d'encodeur de composition avec les propriétés spécifiées.

```
aws ivs-realtime create-encoder-configuration \
    --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

Sortie :

```
{
    "encoderConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABabCDcdEFef",
```

```
        "name": "test-ec",
        "tags": {},
        "video": {
            "bitrate": 3500000,
            "framerate": 30,
            "height": 1080,
            "width": 1920
        }
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous CreateEncoderConfigurationà la section Référence des AWS CLI commandes.

`create-ingest-configuration`

L'exemple de code suivant montre comment utiliser`create-ingest-configuration`.

AWS CLI

Pour créer une configuration d'ingestion

L'`create-ingest-configuration`exemple suivant crée une configuration d'ingestion à l'aide du protocole RTMPS.

```
aws ivs-realtime create-ingest-configuration \
    --name ingest1 \
    --ingest-protocol rtmps
```

Sortie :

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
```

```
        "stageArn": "",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Stream Ingest | Real-Time Streaming dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous CreateIngestConfigurationà la section Référence des AWS CLI commandes.

`create-participant-token`

L'exemple de code suivant montre comment utiliser`create-participant-token`.

AWS CLI

Pour créer un jeton de participant à une étape

L'`create-participant-token`exemple suivant crée un jeton de participation pour l'étape spécifiée.

```
aws ivs-realtime create-participant-token \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --user-id bob
```

Sortie :

```
{
    "participantToken": {
        "expirationTime": "2023-03-07T09:47:43+00:00",
        "participantId": "ABCDEfghij01234KLMN6789",
        "token": "abcd1234defg5678"
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous CreateParticipantTokenà la section Référence des AWS CLI commandes.

`create-stage`

L'exemple de code suivant montre comment utiliser`create-stage`.

AWS CLI

Exemple 1 : pour créer une scène

L'`create-stage`exemple suivant crée un jeton de stage et un jeton de participant à l'étape pour un utilisateur spécifié.

```
aws ivs-realtime create-stage \
    --name stage1 \
    --participant-token-configurations userId=alice
```

Sortie :

```
{
    "participantTokens": [
        {
            "participantId": "ABCDEfghij01234KLMN5678",
            "token": "a1b2c3d4567890ab",
            "userId": "alice"
        }
    ],
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
```

```
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 2 : pour créer une scène et configurer l'enregistrement individuel des participants

L'`create-stage`exemple suivant crée une scène et configure l'enregistrement individuel des participants.

```
aws ivs-realtime create-stage \
    --name stage1 \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh",
 "recordingReconnectWindowSeconds": 100, \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}'
```

Sortie :

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
```

```
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh"
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 3 : pour créer une scène et configurer l'enregistrement individuel des participants avec activation de l'enregistrement par vignettes

L'`create-stage`exemple suivant crée une scène et configure l'enregistrement individuel des participants avec l'enregistrement des vignettes activé.

```
aws ivs-realtime create-stage \
    --name stage1 \
```

```
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh", \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}}'
```

Sortie :

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

- Pour plus de détails sur l'API, reportez-vous CreateStageà la section Référence des AWS CLI commandes.

`create-storage-configuration`

L'exemple de code suivant montre comment utiliser`create-storage-configuration`.

AWS CLI

Pour créer une configuration de stockage de compositions

L'`create-storage-configuration`exemple suivant crée une configuration de stockage de compositions avec les propriétés spécifiées.

```
aws ivs-realtime create-storage-configuration \
    --name "test-sc" --s3 "bucketName=amzn-s3-demo-bucket"
```

Sortie :

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef",
        "name": "test-sc",
        "s3": {
            "bucketName": "amzn-s3-demo-bucket"
        },
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous CreateStorageConfigurationà la section Référence des AWS CLI commandes.

`delete-encoder-configuration`

L'exemple de code suivant montre comment utiliser`delete-encoder-configuration`.

AWS CLI

Pour supprimer la configuration d'un encodeur de composition

Ce qui suit `delete-encoder-configuration` supprime la configuration de l'encodeur de composition spécifiée par l'ARN (Amazon Resource Name) donné.

```
aws ivs-realtime delete-encoder-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABabCDcdEFef"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous DeleteEncoderConfigurationà la section Référence des AWS CLI commandes.

`delete-ingest-configuration`

L'exemple de code suivant montre comment utiliser`delete-ingest-configuration`.

AWS CLI

Exemple 1 : pour supprimer une configuration d'ingestion inactive

L'`delete-ingest-configuration`exemple suivant supprime la configuration d'ingestion inactive pour un ARN de configuration d'ingestion spécifié (Amazon Resource Name).

```
aws ivs-realtime delete-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez IVS Stream Ingest | Real-Time Streaming dans le guide de l'utilisateur d'Amazon Interactive Video Service.

Exemple 2 : Pour forcer la suppression d'une configuration d'ingestion active

L'`delete-ingest-configuration`exemple suivant force la suppression de la configuration d'ingestion active pour un ARN de configuration d'ingestion spécifié (Amazon Resource Name).

```
aws ivs-realtime delete-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234 \
    --force
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez IVS Stream Ingest | Real-Time Streaming dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous DeleteIngestConfigurationà la section Référence des AWS CLI commandes.

`delete-public-key`

L'exemple de code suivant montre comment utiliser`delete-public-key`.

AWS CLI

Pour supprimer une clé publique

Ce qui suit `delete-public-key` supprime la clé publique spécifiée.

```
aws ivs-realtime delete-public-key \
    --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Distribuer des jetons aux participants dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous DeletePublicKeyà la section Référence des AWS CLI commandes.

`delete-stage`

L'exemple de code suivant montre comment utiliser`delete-stage`.

AWS CLI

Pour supprimer une étape

L'`delete-stage`exemple suivant supprime l'étape spécifiée.

```
aws ivs-realtime delete-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous DeleteStageà la section Référence des AWS CLI commandes.

`delete-storage-configuration`

L'exemple de code suivant montre comment utiliser`delete-storage-configuration`.

AWS CLI

Pour supprimer une configuration de stockage de compositions

Ce qui suit `delete-storage-configuration` supprime la configuration de stockage de composition spécifiée par l'ARN (Amazon Resource Name) donné.

```
aws ivs-realtime delete-storage-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous DeleteStorageConfigurationà la section Référence des AWS CLI commandes.

`disconnect-participant`

L'exemple de code suivant montre comment utiliser`disconnect-participant`.

AWS CLI

Pour déconnecter un participant à l'étape

L'`disconnect-participant`exemple suivant déconnecte le participant spécifié de l'étape spécifiée.

```
aws ivs-realtime disconnect-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --participant-id ABCDEfghij01234KLMN5678
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous DisconnectParticipantà la section Référence des AWS CLI commandes.

`get-composition`

L'exemple de code suivant montre comment utiliser`get-composition`.

AWS CLI

Exemple 1 : Pour obtenir une composition avec les paramètres de mise en page par défaut

L'`get-composition`exemple suivant obtient la composition de l'ARN (Amazon Resource Name) spécifié.

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Sortie :

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
```

```
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
            }
        ],
        "layout": {
```

```
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Composite Recording | Real-Time Streaming dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

Exemple 2 : Pour obtenir une composition avec une mise en page PiP

L'`get-composition`exemple suivant obtient la composition de l'ARN (Amazon Resource Name) spécifié, qui utilise la mise en page PiP.

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"
```

Sortie :

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
```

```
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
            }
        ],
        "layout": {
            "pip": {
                "featuredParticipantAttribute": "abcdefg",
                "gridGap": 0,
                "omitStoppedVideo": false,
                "pipBehavior": "STATIC",
                "pipOffset": 0,
                "pipParticipantAttribute": "",
                "pipPosition": "BOTTOM_RIGHT",
                "videoFillMode": "COVER"
            }
        },
```

```
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Composite Recording | Real-Time Streaming dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

Exemple 3 : Pour obtenir une composition avec l'enregistrement de vignettes activé

L'`get-composition`exemple suivant obtient la composition de l'ARN (Amazon Resource Name) spécifié, pour lequel l'enregistrement des vignettes est activé avec les paramètres par défaut.

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Sortie :

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
```

```
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                        "thumbnailConfigurations": [
                           {
                              "targetIntervalSeconds": 60,
                              "storage": [
                                  "SEQUENTIAL"
                              ],
                           }
                        ]
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""               }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
```

```
    }
}
```

Pour plus d'informations, consultez IVS Composite Recording | Real-Time Streaming dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous GetCompositionà la section Référence des AWS CLI commandes.

`get-encoder-configuration`

L'exemple de code suivant montre comment utiliser`get-encoder-configuration`.

AWS CLI

Pour obtenir la configuration d'un encodeur de composition

L'`get-encoder-configuration`exemple suivant obtient la configuration de l'encodeur de composition spécifiée par l'ARN (Amazon Resource Name) donné.

```
aws ivs-realtime get-encoder-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

Sortie :

```
{
    "encoderConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
        "name": "test-ec",
        "tags": {},
        "video": {
            "bitrate": 3500000,
            "framerate": 30,
            "height": 1080,
            "width": 1920
        }
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous GetEncoderConfigurationà la section Référence des AWS CLI commandes.

`get-ingest-configuration`

L'exemple de code suivant montre comment utiliser`get-ingest-configuration`.

AWS CLI

Pour obtenir les informations de configuration d'ingestion

L'`get-ingest-configuration`exemple suivant obtient la configuration d'ingestion pour un ARN de configuration d'ingestion spécifié (Amazon Resource Name).

```
aws ivs-realtime get-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234
```

Sortie :

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Stream Ingest | Real-Time Streaming dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous GetIngestConfigurationà la section Référence des AWS CLI commandes.

`get-participant`

L'exemple de code suivant montre comment utiliser`get-participant`.

AWS CLI

Exemple 1 : Pour recruter un participant à l'étape

L'`get-participant`exemple suivant obtient le participant à l'étape pour un ID de participant et un identifiant de session spécifiés dans l'ARN de l'étape (Amazon Resource Name) spécifié.

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "participant": {
        "browserName": "Google Chrome",
        "browserVersion": "116",
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName": "Comcast",
        "osName": "Microsoft Windows 10 Pro",
        "osVersion": "10.0.19044",
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 2 : pour obtenir un participant à une étape qui a été répliqué sur une autre étape

L'`get-participant`exemple suivant obtient le participant à l'étape pour un ID de participant et un identifiant de session spécifiés dans l'ARN de l'étape (Amazon Resource Name) spécifié, lorsque le participant a également été répliqué sur une autre étape.

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "participant": {
        "browserName": "Google Chrome",
        "browserVersion": "116",
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName": "Comcast",
        "osName": "Microsoft Windows 10 Pro",
        "osVersion": "10.0.19044",
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "replicationState": "ACTIVE",
        "replicationType": "SOURCE",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 3 : Pour obtenir un participant à une étape qui a été dupliqué depuis une autre étape

L'`get-participant`exemple suivant obtient le participant à l'étape pour un ID de participant et un identifiant de session spécifiés dans l'ARN de l'étape (Amazon Resource Name) spécifié, lorsque le participant a été répliqué depuis une autre étape.

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "participant": {
        "browserName": "Google Chrome",
        "browserVersion": "116",
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName": "Comcast",
        "osName": "Microsoft Windows 10 Pro",
        "osVersion": "10.0.19044",
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "replicationState": "ACTIVE",
        "replicationType": "REPLICA",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

- Pour plus de détails sur l'API, reportez-vous GetParticipantà la section Référence des AWS CLI commandes.

`get-public-key`

L'exemple de code suivant montre comment utiliser`get-public-key`.

AWS CLI

Pour obtenir une clé publique existante utilisée pour signer les jetons des participants à la scène

L'`get-public-key`exemple suivant obtient une clé publique spécifiée par l'ARN fourni, pour signer les jetons des participants à l'étape.

```
aws ivs-realtime get-public-key \
    --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

Sortie :

```
{
    "publicKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
        "name": "",
        "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcFOOAzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
        "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Distribuer des jetons aux participants dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous GetPublicKeyà la section Référence des AWS CLI commandes.

`get-stage-session`

L'exemple de code suivant montre comment utiliser`get-stage-session`.

AWS CLI

Pour obtenir une séance sur scène

L'`get-stage-session`exemple suivant obtient la session d'étape pour un ID de session spécifié d'un ARN d'étape spécifié (Amazon Resource Name).

```
aws ivs-realtime get-stage-session \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

Sortie :

```
{
    "stageSession": {
        "endTime": "2023-04-26T20:36:29+00:00",
        "sessionId": "st-a1b2c3d4e5f6g",
        "startTime": "2023-04-26T20:30:29.602000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous GetStageSessionà la section Référence des AWS CLI commandes.

`get-stage`

L'exemple de code suivant montre comment utiliser`get-stage`.

AWS CLI

Pour obtenir les informations de configuration d'une étape

L'`get-stage`exemple suivant obtient la configuration de stage pour un ARN d'étape spécifié (Amazon Resource Name).

```
aws ivs-realtime get-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Sortie :

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
```

```
            "storageConfigurationArn": "",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "test",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

- Pour plus de détails sur l'API, reportez-vous GetStageà la section Référence des AWS CLI commandes.

`get-storage-configuration`

L'exemple de code suivant montre comment utiliser`get-storage-configuration`.

AWS CLI

Pour obtenir une configuration de stockage de compositions

L'`get-storage-configuration`exemple suivant obtient la configuration de stockage de composition spécifiée par l'ARN (Amazon Resource Name) donné.

```
aws ivs-realtime get-storage-configuration \
    --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh"
```

Sortie :

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
        "name": "test-sc",
        "s3": {
            "bucketName": "amzn-s3-demo-bucket"
        },
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous GetStorageConfigurationà la section Référence des AWS CLI commandes.

`import-public-key`

L'exemple de code suivant montre comment utiliser`import-public-key`.

AWS CLI

Pour importer une clé publique existante à utiliser pour signer les jetons des participants à l'étape

L'`import-public-key`exemple suivant importe une clé publique à partir d'un fichier matériel, à utiliser pour signer les jetons des participants à l'étape.

```
aws ivs-realtime import-public-key \
    --public-key-material="`cat public.pem`"
```

Sortie :

```
{
    "publicKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
        "name": "",
        "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcFOOAzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
        "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Distribuer des jetons aux participants dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous ImportPublicKeyà la section Référence des AWS CLI commandes.

`list-compositions`

L'exemple de code suivant montre comment utiliser`list-compositions`.

AWS CLI

Pour obtenir une liste de compositions

Vous trouverez ci-dessous la `list-compositions` liste de toutes les compositions pour votre AWS compte, dans la AWS région où la demande d'API est traitée.

```
aws ivs-realtime list-compositions
```

Sortie :

```
{
    "compositions": [
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
abcdABCDefgh",
            "destinations": [
```

```
                {
                    "id": "AabBCcdDEefF",
                    "startTime": "2023-10-16T23:25:23+00:00",
                    "state": "ACTIVE"
                }
            ],
            "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
defgABCDabcd",
            "startTime": "2023-10-16T23:25:21+00:00",
            "state": "ACTIVE",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
ABcdabCDefgh",
            "destinations": [
                {
                    "endTime": "2023-10-16T23:25:00.786512+00:00",
                    "id": "aABbcCDdeEFf",
                    "startTime": "2023-10-16T23:24:01+00:00",
                    "state": "STOPPED"
                },
                {
                    "endTime": "2023-10-16T23:25:00.786512+00:00",
                    "id": "deEFfaABbcCD",
                    "startTime": "2023-10-16T23:24:01+00:00",
                    "state": "STOPPED"
                }
            ],
            "endTime": "2023-10-16T23:25:00+00:00",
            "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
efghabcdABCD",
            "startTime": "2023-10-16T23:24:00+00:00",
            "state": "STOPPED",
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous ListCompositionsà la section Référence des AWS CLI commandes.

`list-encoder-configurations`

L'exemple de code suivant montre comment utiliser`list-encoder-configurations`.

AWS CLI

Pour répertorier les configurations des encodeurs de composition

Vous trouverez ci-dessous la `list-encoder-configurations` liste de toutes les configurations d'encodeur de composition pour votre AWS compte, dans la AWS région où la demande d'API est traitée.

```
aws ivs-realtime list-encoder-configurations
```

Sortie :

```
{
    "encoderConfigurations": [
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
            "name": "test-ec-1",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABCefgEFGabc",
            "name": "test-ec-2",
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous ListEncoderConfigurationsà la section Référence des AWS CLI commandes.

`list-ingest-configurations`

L'exemple de code suivant montre comment utiliser`list-ingest-configurations`.

AWS CLI

Pour obtenir des informations récapitulatives sur toutes les configurations d'ingestion

L'`list-ingest-configurations`exemple suivant répertorie toutes les configurations d'ingestion pour votre AWS compte, dans la AWS région où la demande d'API est traitée.

```
aws ivs-realtime list-ingest-configurations
```

Sortie :

```
{
    "ingestConfigurations": [
        {
            "name": "",
            "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
XYZuvwSt4567",
            "ingestProtocol": "RTMPS",
            "stageArn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
            "participnatId": "abC789Xyz456",
            "state": "INACTIVE"
            "userId": "",
        }
    ]
}
```

Pour plus d'informations, consultez IVS Stream Ingest | Real-Time Streaming dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous ListIngestConfigurationsà la section Référence des AWS CLI commandes.

`list-participant-events`

L'exemple de code suivant montre comment utiliser`list-participant-events`.

AWS CLI

Exemple 1 : Pour obtenir une liste des événements organisés par les participants à la scène

L'`list-participant-events`exemple suivant répertorie tous les événements participants pour un ID de participant et un ID de session spécifiés pour un ARN (Amazon Resource Name) d'étape spécifié.

```
aws ivs-realtime list-participant-events \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "events": [
        {
            "eventTime": "2023-04-26T20:36:28+00:00",
            "name": "LEFT",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2023-04-26T20:36:28+00:00",
            "name": "PUBLISH_STOPPED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2023-04-26T20:30:34+00:00",
            "name": "JOINED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2023-04-26T20:30:34+00:00",
            "name": "PUBLISH_STARTED",
            "participantId": "abCDEf12GHIj"
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 2 : pour obtenir une liste des événements des participants à l'étape, y compris l'arrêt et le démarrage de la réplication des participants

L'`list-participant-events`exemple suivant répertorie tous les événements des participants pour un ID de session spécifié ou un ARN d'étape spécifique (Amazon Resource Name), où un participant est répliqué vers une autre étape.

```
aws ivs-realtime list-participant-events \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "events": [
        {
            "eventTime": "2025-04-26T20:36:28+00:00",
            "name": "LEFT",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:36:28+00:00",
            "name": "PUBLISH_STOPPED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:30:34+00:00",
            "name": "JOINED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:30:34+00:00",
            "name": "PUBLISH_STARTED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "name": "REPLICATION_STARTED",
            "participantId": "abCDEf12GHIj",
            "eventTime": "2025-04-26T20:30:34+00:00",
            "destinationStageArn": "arn:aws:ivs:us-west-2:12345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a"
        },
        {
            "name": "REPLICATION_STOPPED",
```

```
            "participantId": "abCDEf12GHIj",
            "eventTime": "2025-04-26T20:32:34+00:00",
            "destinationStageArn": "arn:aws:ivs:us-west-2:12345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a"
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

- Pour plus de détails sur l'API, reportez-vous ListParticipantEventsà la section Référence des AWS CLI commandes.

`list-participant-replicas`

L'exemple de code suivant montre comment utiliser`list-participant-replicas`.

AWS CLI

Pour obtenir une liste des répliques des participants à la scène

L'`list-participant-replicas`exemple suivant répertorie tous les participants à l'étape répliqués depuis l'ARN de l'étape source spécifié (Amazon Resource Name) vers une autre étape.

```
aws ivs-realtime list-participant-replicas \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "replicas": [
        {
            "sourceStageArn": "arn:aws:ivs:us-west-2:123456789012:stage/
abcdABCDefgh",
            "participantId": "abCDEf12GHIj",
            "sourceSessionId": "st-a1b2c3d4e5f6g",
            "destinationStageArn": "arn:aws:ivs:us-west-2:012345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a",
            "replicationState": "ACTIVE"
```

```
        }
    ]
}
```

Pour plus d'informations, consultez IVS Participant Replication dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous ListParticipantReplicasà la section Référence des AWS CLI commandes.

`list-participants`

L'exemple de code suivant montre comment utiliser`list-participants`.

AWS CLI

Exemple 1 : Pour obtenir la liste des participants à l'étape

L'`list-participants`exemple suivant répertorie tous les participants pour un ID de session spécifié d'un ARN d'étape spécifié (Amazon Resource Name).

```
aws ivs-realtime list-participants \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

Sortie :

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj",
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": ""
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 2 : Pour obtenir une liste des participants à une étape, lorsqu'un participant a été répliqué sur une autre étape

L'`list-participants`exemple suivant répertorie tous les participants pour un ID de session spécifié ou un ARN d'étape spécifique (Amazon Resource Name), lorsqu'un participant a été répliqué vers une autre étape.

```
aws ivs-realtime list-participants \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

Sortie :

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj",
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": "",
            "replicationState": "ACTIVE",
            "replicationType": "SOURCE",
            "sourceStageArn": "",
            "sourceSessionId": ""
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 3 : Pour obtenir une liste des participants à une étape, lorsqu'un participant a été dupliqué depuis une autre étape

L'`list-participants`exemple suivant répertorie tous les participants pour un ID de session spécifié ou un ARN d'étape spécifique (Amazon Resource Name), lorsqu'un participant a été répliqué depuis une autre étape.

```
aws ivs-realtime list-participants \
```

```
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

Sortie :

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj",
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": "",
            "replicationState": "ACTIVE",
            "replicationType": "REPLICA",
            "sourceStageArn": "arn:aws:ivs:us-west-2:123456789012:stage/
abcdABCDefgh",
            "sourceSessionId": "st-a1b2c3d4e5f6g"
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

- Pour plus de détails sur l'API, reportez-vous ListParticipantsà la section Référence des AWS CLI commandes.

`list-public-keys`

L'exemple de code suivant montre comment utiliser`list-public-keys`.

AWS CLI

Pour répertorier les clés publiques existantes disponibles pour signer les jetons des participants

L'`list-public-keys`exemple suivant répertorie toutes les clés publiques disponibles pour signer les jetons des participants à l'étape, dans la AWS région où la demande d'API est traitée.

```
aws ivs-realtime list-public-keys
```

Sortie :

```
{
    "publicKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
            "name": "",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/3bcdABCDefg4",
            "name": "",
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Distribuer des jetons aux participants dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous ListPublicKeysà la section Référence des AWS CLI commandes.

`list-stage-sessions`

L'exemple de code suivant montre comment utiliser`list-stage-sessions`.

AWS CLI

Pour obtenir la liste des sessions sur scène

L'`list-stage-sessions`exemple suivant répertorie toutes les sessions pour un ARN d'étape spécifié (Amazon Resource Name).

```
aws ivs-realtime list-stage-sessions \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Sortie :

```
{
    "stageSessions": [
```

```
        {
            "endTime": "2023-04-26T20:36:29+00:00",
            "sessionId": "st-a1b2c3d4e5f6g",
            "startTime": "2023-04-26T20:30:29.602000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous ListStageSessionsà la section Référence des AWS CLI commandes.

`list-stages`

L'exemple de code suivant montre comment utiliser`list-stages`.

AWS CLI

Pour obtenir des informations récapitulatives sur toutes les étapes

L'`list-stages`exemple suivant répertorie toutes les étapes de votre AWS compte, dans la AWS région où la demande d'API est traitée.

```
aws ivs-realtime list-stages
```

Sortie :

```
{
    "stages": [
        {
            "activeSessionId": "st-a1b2c3d4e5f6g",
            "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
            "name": "stage1",
            "tags": {}
        },
        {
            "activeSessionId": "st-a123bcd456efg",
            "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcd1234ABCD",
            "name": "stage2",
            "tags": {}
```

```
        },
        {
            "activeSessionId": "st-abcDEF1234ghi",
            "arn": "arn:aws:ivs:us-west-2:123456789012:stage/ABCD1234efgh",
            "name": "stage3",
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous ListStagesà la section Référence des AWS CLI commandes.

`list-storage-configurations`

L'exemple de code suivant montre comment utiliser`list-storage-configurations`.

AWS CLI

Pour répertorier les configurations de stockage de compositions

Vous trouverez ci-dessous la `list-storage-configurations` liste de toutes les configurations de stockage des compositions pour votre AWS compte, dans la AWS région où la demande d'API est traitée.

```
aws ivs-realtime list-storage-configurations
```

Sortie :

```
{
    "storageConfigurations": [
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
            "name": "test-sc-1",
            "s3": {
                "bucketName": "amzn-s3-demo-bucket-1"
            },
            "tags": {}
```

```
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABCefgEFGabc",
            "name": "test-sc-2",
            "s3": {
                "bucketName": "amzn-s3-demo-bucket-2"
            },
            "tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous ListStorageConfigurationsà la section Référence des AWS CLI commandes.

`start-composition`

L'exemple de code suivant montre comment utiliser`start-composition`.

AWS CLI

Exemple 1 : Pour démarrer une composition avec les paramètres de mise en page par défaut

L'`start-composition`exemple suivant lance une composition pour la scène spécifiée à diffuser vers les emplacements spécifiés.

```
aws ivs-realtime start-composition \
    --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
    --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
        "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
        {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
        "recordingConfiguration": {"hlsConfiguration":
 {"targetSegmentDurationSeconds": 5}}, \
        "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}}]'
```

Sortie :


```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "state": "STARTING"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 5
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
```

```
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "STARTING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Composite Recording | Real-Time Streaming dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

Exemple 2 : Pour démarrer une composition avec une mise en page PiP

L'`start-composition`exemple suivant démarre une composition pour la scène spécifiée à diffuser vers les emplacements spécifiés à l'aide de la mise en page PiP.

```
aws ivs-realtime start-composition \
    --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
    --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
        "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
        {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
        "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}}]' \
    --layout pip='{featuredParticipantAttribute="abcdefg"}'
```

Sortie :

```
{
    "composition": {
```

```
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "state": "STARTING"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "state": "STARTING"
            }
        ],
        "layout": {
```

```
            "pip": {
                "featuredParticipantAttribute": "abcdefg",
                "gridGap": 0,
                "omitStoppedVideo": false,
                "pipBehavior": "STATIC",
                "pipOffset": 0,
                "pipParticipantAttribute": "",
                "pipPosition": "BOTTOM_RIGHT",
                "videoFillMode": "COVER"
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "STARTING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Composite Recording | Real-Time Streaming dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

Exemple 3 : Pour démarrer une composition avec l'enregistrement de vignettes activé

L'`start-composition`exemple suivant démarre une composition pour la scène spécifiée à diffuser vers les emplacements spécifiés avec l'enregistrement des vignettes activé.

```
aws ivs-realtime start-composition \
    --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
    --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
        "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
        {"s3": {"encoderConfigurationArns": ["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
        "storageConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE", \
        "thumbnailConfigurations": [{"storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}]}}]'
```

Sortie :

```
{
    "composition": {
```

```
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "state": "STARTING"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                        "thumbnailConfigurations": [
                           {
                              "targetIntervalSeconds": 60,
                              "storage": [
                                  "SEQUENTIAL"
                              ]
                           }
                        ]
                    }
                },
                "detail": {
                    "s3": {
```

```
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "STARTING",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Composite Recording | Real-Time Streaming dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous StartCompositionà la section Référence des AWS CLI commandes.

`start-participant-replication`

L'exemple de code suivant montre comment utiliser`start-participant-replication`.

AWS CLI

Pour commencer à répliquer un participant d'une étape à une autre

L'`start-participant-replication`exemple suivant réplique un participant d'une étape source vers une étape de destination, chaque étape étant spécifiée par son ARN (Amazon Resource Name).

```
aws ivs-realtime start-participant-replication \
```

```
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --destination-stage-arn arn:aws:ivs:us-west-2:234567890123:stage/bcdABCDefghi \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "accessControlAllowOrigin": "*",
    "accessControlExposeHeaders": "Access-Control-Allow-Origin,Access-Control-
Expose-Headers,Cache-Control,Content-Length, \
    Content-Security-Policy,Content-Type,date,Strict-Transport-Security,x-amz-apigw-
id,x-amzn-errormessage,x-amzn-errortype, \
    x-amzn-requestid,x-amzn-trace-id,X-Content-Type-Options,X-Frame-Options",
    "cacheControl": "no-store, no-cache",
    "contentSecurityPolicy": "default-src 'self'; upgrade-insecure-requests;",
    "strictTransportSecurity": "max-age:47304000; includeSubDomains",
    "xContentTypeOptions": "nosniff",
    "xFrameOptions": "DENY"
}
```

Pour plus d'informations, consultez IVS Participant Replication dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous StartParticipantReplicationà la section Référence des AWS CLI commandes.

`stop-composition`

L'exemple de code suivant montre comment utiliser`stop-composition`.

AWS CLI

Pour arrêter une composition

Ce qui suit `stop-composition` arrête la composition spécifiée par l'ARN (Amazon Resource Name) donné.

```
aws ivs-realtime stop-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous StopCompositionà la section Référence des AWS CLI commandes.

`stop-participant-replication`

L'exemple de code suivant montre comment utiliser`stop-participant-replication`.

AWS CLI

Pour arrêter de répliquer un participant d'une étape à une autre

L'`stop-participant-replication`exemple suivant arrête la réplication d'un participant d'une étape source vers une étape de destination, chaque étape étant spécifiée par son ARN (Amazon Resource Name).

```
aws ivs-realtime stop-participant-replication \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --destination-stage-arn arn:aws:ivs:us-west-2:234567890123:stage/bcdABCDefghi \
    --participant-id abCDEf12GHIj
```

Sortie :

```
{
    "accessControlAllowOrigin": "*",
    "accessControlExposeHeaders": "Access-Control-Allow-Origin,Access-Control-
Expose-Headers,Cache-Control,Content-Length, \
    Content-Security-Policy,Content-Type,date,Strict-Transport-Security,x-amz-apigw-
id,x-amzn-errormessage,x-amzn-errortype, \
    x-amzn-requestid,x-amzn-trace-id,X-Content-Type-Options,X-Frame-Options",
    "cacheControl": "no-store, no-cache",
    "contentSecurityPolicy": "default-src 'self'; upgrade-insecure-requests;",
    "strictTransportSecurity": "max-age:47304000; includeSubDomains",
    "xContentTypeOptions": "nosniff",
    "xFrameOptions": "DENY"
}
```

Pour plus d'informations, consultez IVS Participant Replication dans le guide de l'utilisateur d'Amazon IVS Real-Time Streaming.

- Pour plus de détails sur l'API, reportez-vous StopParticipantReplicationà la section Référence des AWS CLI commandes.

`update-ingest-configuration`

L'exemple de code suivant montre comment utiliser`update-ingest-configuration`.

AWS CLI

Pour mettre à jour une configuration d'ingestion

L'`update-inegst-configuration`exemple suivant met à jour une configuration d'ingestion pour l'associer à une étape.

```
aws ivs-realtime update-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234 \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Sortie :

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez IVS Stream Ingest | Real-Time Streaming dans le guide de l'utilisateur d'Amazon Interactive Video Service.

- Pour plus de détails sur l'API, reportez-vous UpdateIngestConfigurationà la section Référence des AWS CLI commandes.

update-stage

L'exemple de code suivant montre comment utiliserupdate-stage.

AWS CLI

Exemple 1 : pour mettre à jour la configuration d'une étape

L'update-stageexemple suivant met à jour une étape pour un ARN d'étape spécifié afin de mettre à jour le nom de l'étape et de configurer l'enregistrement individuel des participants avec l'enregistrement des vignettes activé.

```
aws ivs-realtime update-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh",
 "recordingReconnectWindowSeconds": 100, \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}} \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}' \
    --name stage1a
```

Sortie :

```
{
    "stage": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
```

```
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

Exemple 2 : pour mettre à jour la configuration d'une étape, notamment en désactivant l'enregistrement des répliques des participants

L'`update-stage`exemple suivant met à jour un stage pour un ARN d'étape spécifié afin de mettre à jour le nom de l'étape et de configurer l'enregistrement individuel des participants avec l'enregistrement des vignettes activé et l'enregistrement des répliques des participants désactivé.

```
aws ivs-realtime update-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh",
 "recordingReconnectWindowSeconds": 100, \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}, "recordParticipantReplicas":false} \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}' \
    --name stage1a
```

Sortie :

```
{
    "stage": {
```

```
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": false
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

Pour plus d'informations, consultez la section Activation de plusieurs hôtes sur un flux Amazon IVS dans le guide de l'utilisateur du streaming à faible latence d'Amazon IVS.

- Pour plus de détails sur l'API, reportez-vous UpdateStageà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Kendra AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Kendra.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-data-source`

L'exemple de code suivant montre comment utiliser`create-data-source`.

AWS CLI

Pour créer un connecteur de source de données Amazon Kendra

Ce qui suit `create-data-source` permet de créer et de configurer un connecteur de source de données Amazon Kendra. Vous pouvez l'utiliser `describe-data-source` pour afficher l'état d'un connecteur de source de données et lire tout message d'erreur s'il indique que la création complète d'un connecteur de source de données a « ÉCHOUÉ ».

```
aws kendra create-data-source \
    --name "example data source 1" \
    --description "Example data source 1 for example index 1 contains the first set
 of example documents" \
    --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",
 "Value": "aws"}' \
    --role-arn "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource" \
    --index-id exampleindex1 \
    --language-code "es" \
    --schedule "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *" \
```

```
    --configuration '{"TemplateConfiguration": {"Template": file://
s3schemaconfig.json}}' \
    --type "TEMPLATE" \
    --custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":
 {"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
 "S3Bucket": "s3://amzn-s3-demo-bucket/scanned-image-text-example-docs"}, "RoleArn":
 "arn:aws:iam:my-account-id:role/KendraRoleForCDE"}' \
    --vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":
 ["subnet-1c234","subnet-2b134"]}'
```

Sortie :

```
{
    "Id": "exampledatasource1"
}
```

Pour plus d'informations, consultez Getting started with a Amazon Kendra index and data source connector dans le guide du développeur Amazon Kendra.

- Pour plus de détails sur l'API, reportez-vous CreateDataSourceà la section Référence des AWS CLI commandes.

`create-index`

L'exemple de code suivant montre comment utiliser`create-index`.

AWS CLI

Pour créer un index Amazon Kendra

Ce qui suit `create-index` crée et configure un index Amazon Kendra. Vous pouvez l'utiliser `describe-index` pour afficher l'état d'un index et lire tout message d'erreur s'il indique qu'un index a « FAILED » à créer complètement.

```
aws kendra create-index \
    --name "example index 1" \
    --description "Example index 1 contains the first set of example documents" \
    --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",
 "Value": "aws"}' \
    --role-arn "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex" \
    --edition "DEVELOPER_EDITION" \
    --server-side-encryption-configuration '{"KmsKeyId": "my-kms-key-id"}' \
```

```
    --user-context-policy "USER_TOKEN" \
    --user-token-configurations '{"JsonTokenTypeConfiguration":
 {"GroupAttributeField": "groupNameField", "UserNameAttributeField":
 "userNameField"}}'
```

Sortie :

```
{
   "Id": index1
}
```

Pour plus d'informations, consultez Getting started with a Amazon Kendra index and data source connector dans le guide du développeur Amazon Kendra.

- Pour plus de détails sur l'API, reportez-vous CreateIndexà la section Référence des AWS CLI commandes.

`describe-data-source`

L'exemple de code suivant montre comment utiliser`describe-data-source`.

AWS CLI

Pour obtenir des informations sur un connecteur de source de données Amazon Kendra

Vous trouverez ci-dessous `describe-data-source` des informations sur un connecteur de source de données Amazon Kendra. Vous pouvez consulter la configuration d'un connecteur de source de données et lire tous les messages d'erreur si l'état indique que la création complète d'un connecteur de source de données a « ÉCHOUÉ ».

```
aws kendra describe-data-source \
    --id exampledatasource1 \
    --index-id exampleindex1
```

Sortie :

```
{
    "Configuration": {
        "TemplateConfiguration": {
            "Template": {
                "connectionConfiguration": {
                    "repositoryEndpointMetadata": {
```

```
                    "BucketName": "amzn-s3-demo-bucket"
                }
            },
            "repositoryConfigurations": {
                "document":{
                    "fieldMappings": [
                        {
                            "indexFieldName":"_document_title",
                            "indexFieldType":"STRING",
                            "dataSourceFieldName": "title"
                        },
                        {
                            "indexFieldName":"_last_updated_at",
                            "indexFieldType":"DATE",
                            "dataSourceFieldName": "modified_date"
                        }
                    ]
                }
            },
            "additionalProperties": {
                "inclusionPatterns": [
                    "*.txt",
                    "*.doc",
                    "*.docx"
                ],
                "exclusionPatterns": [
                    "*.json"
                ],
                "inclusionPrefixes": [
                    "PublicExampleDocsFolder"
                ],
                "exclusionPrefixes": [
                    "PrivateDocsFolder/private"
                ],
                "aclConfigurationFilePath": "ExampleDocsFolder/AclConfig.json",
                "metadataFilesPrefix": "metadata"
            },
            "syncMode": "FULL_CRAWL",
            "type" : "S3",
            "version": "1.0.0"
        }
    }
},
"CreatedAt": 2024-02-25T13:30:10+00:00,
```

```
    "CustomDocumentEnrichmentConfiguration": {
        "PostExtractionHookConfiguration": {
            "LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
            "S3Bucket": "s3://amzn-s3-demo-bucket/scanned-image-text-example-docs/
function"
        },
        "RoleArn": "arn:aws:iam:my-account-id:role/KendraRoleForCDE"
    }
    "Description": "Example data source 1 for example index 1 contains the first set
 of example documents",
    "Id": exampledatasource1,
    "IndexId": exampleindex1,
    "LanguageCode": "en",
    "Name": "example data source 1",
    "RoleArn": "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource",
    "Schedule": "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *",
    "Status": "ACTIVE",
    "Type": "TEMPLATE",
    "UpdatedAt": 1709163615,
    "VpcConfiguration": {
        "SecurityGroupIds": ["sg-1234567890abcdef0"],
        "SubnetIds": ["subnet-1c234","subnet-2b134"]
    }
}
```

Pour plus d'informations, consultez Getting started with a Amazon Kendra index and data source connector dans le guide du développeur Amazon Kendra.

- Pour plus de détails sur l'API, reportez-vous DescribeDataSourceà la section Référence des AWS CLI commandes.

`describe-index`

L'exemple de code suivant montre comment utiliser`describe-index`.

AWS CLI

Pour obtenir des informations sur un index Amazon Kendra

Vous trouverez ci-dessous `describe-index` des informations sur un index Amazon Kendra. Vous pouvez consulter la configuration d'un index et lire tous les messages d'erreur si l'état indique qu'un index a « FAILED » à créer complètement.

```
aws kendra describe-index \
    --id exampleindex1
```

Sortie :

```
{
    "CapacityUnits": {
        "QueryCapacityUnits": 0,
        "StorageCapacityUnits": 0
    },
    "CreatedAt": 2024-02-25T12:30:10+00:00,
    "Description": "Example index 1 contains the first set of example documents",
    "DocumentMetadataConfigurations": [
        {
            "Name": "_document_title",
            "Relevance": {
                "Importance": 8
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": false
            },
            "Type": "STRING_VALUE"
        },
        {
            "Name": "_document_body",
            "Relevance": {
                "Importance": 5
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": false
            },
            "Type": "STRING_VALUE"
        },
        {
            "Name": "_last_updated_at",
            "Relevance": {
                "Importance": 6,
```

```
                "Duration": "2628000s",
                "Freshness": true
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": true
            },
            "Type": "DATE_VALUE"
        },
        {
            "Name": "department_custom_field",
            "Relevance": {
                "Importance": 7,
                "ValueImportanceMap": {
                    "Human Resources" : 4,
                    "Marketing and Sales" : 2,
                    "Research and innvoation" : 3,
                    "Admin" : 1
                }
            },
            "Search": {
                "Displayable": true,
                "Facetable": true,
                "Searchable": true,
                "Sortable": true
            },
            "Type": "STRING_VALUE"
        }
    ],
    "Edition": "DEVELOPER_EDITION",
    "Id": "index1",
    "IndexStatistics": {
        "FaqStatistics": {
            "IndexedQuestionAnswersCount": 10
        },
        "TextDocumentStatistics": {
            "IndexedTextBytes": 1073741824,
            "IndexedTextDocumentsCount": 1200
        }
    },
    "Name": "example index 1",
    "RoleArn": "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex",
```

```
    "ServerSideEncryptionConfiguration": {
        "KmsKeyId": "my-kms-key-id"
    },
    "Status": "ACTIVE",
    "UpdatedAt": 1709163615,
    "UserContextPolicy": "USER_TOKEN",
    "UserTokenConfigurations": [
        {
            "JsonTokenTypeConfiguration": {
                "GroupAttributeField": "groupNameField",
                "UserNameAttributeField": "userNameField"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with a Amazon Kendra index and data source connector dans le guide du développeur Amazon Kendra.

- Pour plus de détails sur l'API, reportez-vous DescribeIndexà la section Référence des AWS CLI commandes.

`update-data-source`

L'exemple de code suivant montre comment utiliser`update-data-source`.

AWS CLI

Pour mettre à jour un connecteur de source de données Amazon Kendra

Ce qui suit `update-data-source` met à jour la configuration d'un connecteur de source de données Amazon Kendra. Si l'action aboutit, le service ne renvoie aucune sortie, soit le code d'état HTTP 200, soit le code de retour 0 de la AWS CLI. Vous pouvez l'utiliser `describe-data-source` pour afficher la configuration et l'état d'un connecteur de source de données.

```
aws kendra update-data-source \
    --id exampledatasource1 \
    --index-id exampleindex1 \
    --name "new name for example data source 1" \
    --description "new description for example data source 1" \
    --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForExampleDataSource \
    --configuration '{"TemplateConfiguration": {"Template": file://
s3schemanewconfig.json}}' \
```

```
    --custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":
{"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
"S3Bucket": "s3://amzn-s3-demo-bucket/scanned-image-text-example-docs"}, "RoleArn":
"arn:aws:iam:my-account-id:role/KendraNewRoleForCDE"}' \
    --language-code "es" \
    --schedule "0 0 18 ? * MON,WED,FRI *" \
    --vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":
["subnet-1c234","subnet-2b134"]}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting started with a Amazon Kendra index and data source connector dans le guide du développeur Amazon Kendra.

- Pour plus de détails sur l'API, reportez-vous UpdateDataSourceà la section Référence des AWS CLI commandes.

`update-index`

L'exemple de code suivant montre comment utiliser`update-index`.

AWS CLI

Pour mettre à jour un index Amazon Kendra

Ce qui suit `update-index` met à jour la configuration d'un index Amazon Kendra. Si l'action aboutit, le service ne renvoie aucune sortie, soit le code d'état HTTP 200, soit le code de retour 0 de la AWS CLI. Vous pouvez l'utiliser `describe-index` pour afficher la configuration et le statut d'un index.

```
aws kendra update-index \
    --id enterpriseindex1 \
    --name "new name for Enterprise Edition index 1" \
    --description "new description for Enterprise Edition index 1" \
    --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForEnterpriseIndex \
    --capacity-units '{"QueryCapacityUnits": 2, "StorageCapacityUnits": 1}' \
    --document-metadata-configuration-updates '{"Name": "_document_title",
"Relevance": {"Importance": 6}}, {"Name": "_last_updated_at", "Relevance":
{"Importance": 8}}' \
    --user-context-policy "USER_TOKEN" \
```

```
    --user-token-configurations '{"JsonTokenTypeConfiguration":
{"GroupAttributeField": "groupNameField", "UserNameAttributeField":
"userNameField"}}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting started with a Amazon Kendra index and data source connector dans le guide du développeur Amazon Kendra.

- Pour plus de détails sur l'API, reportez-vous UpdateIndexà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Kinesis AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Winesis.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-stream`

L'exemple de code suivant montre comment utiliser`add-tags-to-stream`.

AWS CLI

Pour ajouter des balises à un flux de données

L'`add-tags-to-stream`exemple suivant attribue une balise avec la clé `samplekey` et la valeur `example` au flux spécifié.

```
aws kinesis add-tags-to-stream \
```

```
    --stream-name samplestream \
    --tags samplekey=example
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, voir AddTagsToStreamla section Référence des AWS CLI commandes.

`create-stream`

L'exemple de code suivant montre comment utiliser`create-stream`.

AWS CLI

Pour créer un flux de données

L'`create-stream`exemple suivant crée un flux de données nommé samplestream avec 3 partitions.

```
aws kinesis create-stream \
    --stream-name samplestream \
    --shard-count 3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'un flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, voir CreateStreamla section Référence des AWS CLI commandes.

`decrease-stream-retention-period`

L'exemple de code suivant montre comment utiliser`decrease-stream-retention-period`.

AWS CLI

Pour réduire la période de rétention des flux de données

L'`decrease-stream-retention-period`exemple suivant réduit la période de rétention (durée pendant laquelle les enregistrements de données sont accessibles après leur ajout au flux) d'un flux nommé samplestream à 48 heures.

```
aws kinesis decrease-stream-retention-period \
    --stream-name samplestream \
    --retention-period-hours 48
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modification de la période de conservation des données dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, voir DecreaseStreamRetentionPeriodla section Référence des AWS CLI commandes.

`delete-stream`

L'exemple de code suivant montre comment utiliser`delete-stream`.

AWS CLI

Pour supprimer un flux de données

L'`delete-stream`exemple suivant supprime le flux de données spécifié.

```
aws kinesis delete-stream \
    --stream-name samplestream
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, voir DeleteStreamla section Référence des AWS CLI commandes.

`deregister-stream-consumer`

L'exemple de code suivant montre comment utiliser`deregister-stream-consumer`.

AWS CLI

Pour annuler l'enregistrement d'un consommateur de flux de données

L'`deregister-stream-consumer`exemple suivant permet de désenregistrer le consommateur spécifié du flux de données spécifié.

```
aws kinesis deregister-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Developing Consumers with Enhanced Fan-Out Using the Kinesis Data Streams API dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeregisterStreamConsumerà la section Référence des AWS CLI commandes.

`describe-limits`

L'exemple de code suivant montre comment utiliser`describe-limits`.

AWS CLI

Pour décrire les limites des partitions

L'`describe-limits`exemple suivant montre les limites et l'utilisation des partitions pour le AWS compte courant.

```
aws kinesis describe-limits
```

Sortie :

```
{
    "ShardLimit": 500,
    "OpenShardCount": 29
}
```

Pour plus d'informations, consultez la section Resharding a Stream dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeLimitsà la section Référence des AWS CLI commandes.

`describe-stream-consumer`

L'exemple de code suivant montre comment utiliser`describe-stream-consumer`.

AWS CLI

Pour décrire un consommateur de flux de données

L'`describe-stream-consumer`exemple suivant renvoie la description du consommateur spécifié, enregistré avec le flux de données spécifié.

```
aws kinesis describe-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

Sortie :

```
{
    "ConsumerDescription": {
        "ConsumerName": "KinesisConsumerApplication",
        "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
        "ConsumerStatus": "ACTIVE",
        "ConsumerCreationTimestamp": 1572383852.0,
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream"
    }
}
```

Pour plus d'informations, consultez la section Reading data from Amazon Kinesis Data Streams dans le manuel du développeur Amazon Kinesis Data Streams.

- Pour plus de détails sur l'API, reportez-vous DescribeStreamConsumerà la section Référence des AWS CLI commandes.

`describe-stream-summary`

L'exemple de code suivant montre comment utiliser`describe-stream-summary`.

AWS CLI

Pour décrire le résumé d'un flux de données

L'`describe-stream-summary`exemple suivant fournit une description résumée (sans la liste des partitions) du flux de données spécifié.

```
aws kinesis describe-stream-summary \
    --stream-name samplestream
```

Sortie :

```
{
    "StreamDescriptionSummary": {
        "StreamName": "samplestream",
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
        "StreamStatus": "ACTIVE",
        "RetentionPeriodHours": 48,
        "StreamCreationTimestamp": 1572297168.0,
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": []
            }
        ],
        "EncryptionType": "NONE",
        "OpenShardCount": 3,
        "ConsumerCount": 0
    }
}
```

Pour plus d'informations, consultez la section Création et gestion de flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeStreamSummaryà la section Référence des AWS CLI commandes.

`describe-stream`

L'exemple de code suivant montre comment utiliser`describe-stream`.

AWS CLI

Pour décrire un flux de données

L'`describe-stream`exemple suivant renvoie les détails du flux de données spécifié.

```
aws kinesis describe-stream \
    --stream-name samplestream
```

Sortie :

```
{
    "StreamDescription": {
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "StartingHashKey": "0",
                    "EndingHashKey": "113427455640312821154458202477256070484"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716829570364423650249261910734372510605801286533 14"
                }
            },
            {
                "ShardId": "shardId-000000000001",
                "HashKeyRange": {
                    "StartingHashKey": "113427455640312821154458202477256070485",
                    "EndingHashKey": "226854911280625642308916404954512140969"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716829793371875635555493326091555237089416346337 46"
                }
            },
            {
                "ShardId": "shardId-000000000002",
                "HashKeyRange": {
                    "StartingHashKey": "226854911280625642308916404954512140970",
                    "EndingHashKey": "340282366920938463463374607431768211455"
                },
                "SequenceNumberRange": {
```

```
                    "StartingSequenceNumber":
 "49600871683001637932762086172474144873796357303140614178"
                }
            }
        ],
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
        "StreamName": "samplestream",
        "StreamStatus": "ACTIVE",
        "RetentionPeriodHours": 24,
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": []
            }
        ],
        "EncryptionType": "NONE",
        "KeyId": null,
        "StreamCreationTimestamp": 1572297168.0
    }
}
```

Pour plus d'informations, consultez la section Création et gestion de flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeStreamà la section Référence des AWS CLI commandes.

`disable-enhanced-monitoring`

L'exemple de code suivant montre comment utiliser`disable-enhanced-monitoring`.

AWS CLI

Pour désactiver la surveillance améliorée des métriques au niveau des partitions

L'`disable-enhanced-monitoring`exemple suivant désactive la surveillance améliorée du flux de données Kinesis pour les métriques au niveau des partitions.

```
aws kinesis disable-enhanced-monitoring \
    --stream-name samplestream --shard-level-metrics ALL
```

Sortie :

```
{
```

```
    "StreamName": "samplestream",
    "CurrentShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords",
        "IteratorAgeMilliseconds",
        "IncomingRecords",
        "ReadProvisionedThroughputExceeded",
        "WriteProvisionedThroughputExceeded",
        "OutgoingBytes"
    ],
    "DesiredShardLevelMetrics": []
}
```

Pour plus d'informations, consultez la section Surveillance des flux dans Amazon Kinesis Data Streams dans le manuel du développeur Amazon Kinesis Data Streams.

- Pour plus de détails sur l'API, reportez-vous DisableEnhancedMonitoringà la section Référence des AWS CLI commandes.

`enable-enhanced-monitoring`

L'exemple de code suivant montre comment utiliser`enable-enhanced-monitoring`.

AWS CLI

Pour permettre une surveillance améliorée des métriques au niveau des partitions

L'`enable-enhanced-monitoring`exemple suivant permet une surveillance améliorée du flux de données Kinesis pour les métriques au niveau des partitions.

```
aws kinesis enable-enhanced-monitoring \
    --stream-name samplestream \
    --shard-level-metrics ALL
```

Sortie :

```
{
    "StreamName": "samplestream",
    "CurrentShardLevelMetrics": [],
    "DesiredShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords",
        "IteratorAgeMilliseconds",
```

```
        "IncomingRecords",
        "ReadProvisionedThroughputExceeded",
        "WriteProvisionedThroughputExceeded",
        "OutgoingBytes"
    ]
}
```

Pour plus d'informations, consultez la section Surveillance des flux dans Amazon Kinesis Data Streams dans le manuel du développeur Amazon Kinesis Data Streams.

- Pour plus de détails sur l'API, reportez-vous EnableEnhancedMonitoringà la section Référence des AWS CLI commandes.

`get-records`

L'exemple de code suivant montre comment utiliser`get-records`.

AWS CLI

Pour obtenir des enregistrements à partir d'une partition

L'`get-records`exemple suivant obtient des enregistrements de données à partir de la partition d'un flux de données Kinesis à l'aide de l'itérateur de partition spécifié.

```
aws kinesis get-records \
    --shard-iterator AAAAAAAAAAF7/0mWD7IuHj1yGv/
TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYeOZyLZrvzzeOFVf9iN99hUPw/w/
b0YWYeehfNvnf1DYt5XpDJghLKr3DzgznkTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/
GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWGE1
```

Sortie :

```
{
    "Records": [],
    "MillisBehindLatest": 80742000
}
```

Pour plus d'informations, consultez Developing Consumers Using the Kinesis Data Streams API with AWS the SDK for Java dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetRecordsà la section Référence des AWS CLI commandes.

`get-shard-iterator`

L'exemple de code suivant montre comment utiliser`get-shard-iterator`.

AWS CLI

Pour obtenir un itérateur de partition

L'`get-shard-iterator`exemple suivant utilise le type d'itérateur de `AT_SEQUENCE_NUMBER` partition et génère un itérateur de partition pour commencer à lire les enregistrements de données exactement à partir de la position indiquée par le numéro de séquence spécifié.

```
aws kinesis get-shard-iterator \
    --stream-name samplestream \
    --shard-id shardId-000000000001 \
    --shard-iterator-type LATEST
```

Sortie :

```
{
    "ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqgH9FifJ+n48XWTh/
IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy
+lUkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn94O8SqeEqwhigwZxNWxID3Ej7YYYcxQi8Q/fIrCjGAy/
n2r5Z9G864YpWDfN9upNNQAR/iiOWKs"
}
```

Pour plus d'informations, consultez Developing Consumers Using the Kinesis Data Streams API with AWS the SDK for Java dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetShardIteratorà la section Référence des AWS CLI commandes.

`increase-stream-retention-period`

L'exemple de code suivant montre comment utiliser`increase-stream-retention-period`.

AWS CLI

Pour augmenter la durée de conservation des flux de données

L'`increase-stream-retention-period`exemple suivant augmente la durée de conservation (durée pendant laquelle les enregistrements de données sont accessibles après leur ajout au flux) du flux spécifié à 168 heures.

```
aws kinesis increase-stream-retention-period \
    --stream-name samplestream \
    --retention-period-hours 168
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modification de la période de conservation des données dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous IncreaseStreamRetentionPeriodà la section Référence des AWS CLI commandes.

`list-shards`

L'exemple de code suivant montre comment utiliser`list-shards`.

AWS CLI

Pour répertorier les partitions d'un flux de données

L'`list-shards`exemple suivant répertorie toutes les partitions du flux spécifié, en commençant par la partition dont l'ID suit immédiatement la valeur spécifiée `exclusive-start-shard-id` de. `shardId-000000000000`

```
aws kinesis list-shards \
    --stream-name samplestream \
    --exclusive-start-shard-id shardId-000000000000
```

Sortie :

```
{
    "Shards": [
        {
            "ShardId": "shardId-000000000001",
            "HashKeyRange": {
                "StartingHashKey": "113427455640312821154458202477256070485",
                "EndingHashKey": "226854911280625642308916404954512140969"
            },
```

```
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "496008716829793371875635555493326091555237089416346337 46"
            }
        },
        {
            "ShardId": "shardId-000000000002",
            "HashKeyRange": {
                "StartingHashKey": "226854911280625642308916404954512140970",
                "EndingHashKey": "340282366920938463463374607431768211455"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "496008716830016379327620861724741448737963573031406141 78"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Listing Shards dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListShardsà la section Référence des AWS CLI commandes.

`list-streams`

L'exemple de code suivant montre comment utiliser`list-streams`.

AWS CLI

Pour répertorier les flux de données

L'`list-streams`exemple suivant répertorie tous les flux de données actifs dans le compte courant et la région.

```
aws kinesis list-streams
```

Sortie :

```
{
    "StreamNames": [
```

```
        "samplestream",
        "samplestream1"
    ]
}
```

Pour plus d'informations, consultez Listing Streams dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListStreamsà la section Référence des AWS CLI commandes.

`list-tags-for-stream`

L'exemple de code suivant montre comment utiliser`list-tags-for-stream`.

AWS CLI

Pour répertorier les balises d'un flux de données

L'`list-tags-for-stream`exemple suivant répertorie les balises associées au flux de données spécifié.

```
aws kinesis list-tags-for-stream \
    --stream-name samplestream
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "samplekey",
            "Value": "example"
        }
    ],
    "HasMoreTags": false
}
```

Pour plus d'informations, consultez la section Marquage de vos flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForStreamà la section Référence des AWS CLI commandes.

`merge-shards`

L'exemple de code suivant montre comment utiliser`merge-shards`.

AWS CLI

Pour fusionner des partitions

L'`merge-shards`exemple suivant fusionne deux partitions adjacentes avec IDs shardID-000000000000 et shardID-000000000001 dans le flux de données spécifié et les combine en une seule partition.

```
aws kinesis merge-shards \
    --stream-name samplestream \
    --shard-to-merge shardId-000000000000 \
    --adjacent-shard-to-merge shardId-000000000001
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Merging Two Shards dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous MergeShardsà la section Référence des AWS CLI commandes.

`put-record`

L'exemple de code suivant montre comment utiliser`put-record`.

AWS CLI

Pour écrire un enregistrement dans un flux de données

L'`put-record`exemple suivant écrit un seul enregistrement de données dans le flux de données spécifié à l'aide de la clé de partition spécifiée.

```
aws kinesis put-record \
    --stream-name samplestream \
    --data sampledatarecord \
    --partition-key samplepartitionkey
```

Sortie :

```
{
    "ShardId": "shardId-000000000009",
    "SequenceNumber": "496009022733575409159899312569015062438784078352975136 18",
    "EncryptionType": "KMS"
}
```

Pour plus d'informations, consultez la section Developing Producers Using the Amazon Kinesis Data Streams API with AWS the SDK for Java dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutRecordà la section Référence des AWS CLI commandes.

`put-records`

L'exemple de code suivant montre comment utiliser`put-records`.

AWS CLI

Pour écrire plusieurs enregistrements dans un flux de données

L'`put-records`exemple suivant écrit un enregistrement de données à l'aide de la clé de partition spécifiée et un autre enregistrement de données à l'aide d'une clé de partition différente en un seul appel.

```
aws kinesis put-records \
    --stream-name samplestream \
    --
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2
```

Sortie :

```
{
    "FailedRecordCount": 0,
    "Records": [
        {
            "SequenceNumber":
 "49600883331171471519674795588238531498465399900093808706",
            "ShardId": "shardId-000000000004"
```

```
        },
        {
            "SequenceNumber":
 "49600902273357540915989931256902715169698037101720764562",
            "ShardId": "shardId-000000000009"
        }
    ],
    "EncryptionType": "KMS"
}
```

Pour plus d'informations, consultez la section Developing Producers Using the Amazon Kinesis Data Streams API with AWS the SDK for Java dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutRecordsà la section Référence des AWS CLI commandes.

`register-stream-consumer`

L'exemple de code suivant montre comment utiliser`register-stream-consumer`.

AWS CLI

Pour enregistrer un consommateur de flux de données

L'`register-stream-consumer`exemple suivant enregistre un consommateur appelé `KinesisConsumerApplication` avec le flux de données spécifié.

```
aws kinesis register-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

Sortie :

```
{
    "Consumer": {
        "ConsumerName": "KinesisConsumerApplication",
        "ConsumerARN": "arn:aws:kinesis:us-west-2: 123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
        "ConsumerStatus": "CREATING",
        "ConsumerCreationTimestamp": 1572383852.0
    }
```

```
}
```

Pour plus d'informations, consultez Developing Consumers with Enhanced Fan-Out Using the Kinesis Data Streams API dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RegisterStreamConsumerà la section Référence des AWS CLI commandes.

`remove-tags-from-stream`

L'exemple de code suivant montre comment utiliser`remove-tags-from-stream`.

AWS CLI

Pour supprimer des balises d'un flux de données

L'`remove-tags-from-stream`exemple suivant supprime la balise avec la clé spécifiée du flux de données spécifié.

```
aws kinesis remove-tags-from-stream \
    --stream-name samplestream \
    --tag-keys samplekey
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos flux dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RemoveTagsFromStreamà la section Référence des AWS CLI commandes.

`split-shard`

L'exemple de code suivant montre comment utiliser`split-shard`.

AWS CLI

Pour diviser des fragments

L'`split-shard`exemple suivant divise le fragment spécifié en deux nouveaux fragments en utilisant une nouvelle clé de hachage de départ de 10.

```
aws kinesis split-shard \
    --stream-name samplestream \
    --shard-to-split shardId-000000000000 \
    --new-starting-hash-key 10
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Splitting a Shard du manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SplitShardà la section Référence des AWS CLI commandes.

`start-stream-encryption`

L'exemple de code suivant montre comment utiliser`start-stream-encryption`.

AWS CLI

Pour activer le chiffrement des flux de données

L'`start-stream-encryption`exemple suivant active le chiffrement côté serveur pour le flux spécifié, à l'aide de la clé AWS KMS spécifiée.

```
aws kinesis start-stream-encryption \
    --encryption-type KMS \
    --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-
b334-4d3eb496e452 \
    --stream-name samplestream
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Protection des données dans Amazon Kinesis Data Streams dans le manuel du développeur Amazon Kinesis Data Streams.

- Pour plus de détails sur l'API, reportez-vous StartStreamEncryptionà la section Référence des AWS CLI commandes.

`stop-stream-encryption`

L'exemple de code suivant montre comment utiliser`stop-stream-encryption`.

AWS CLI

Pour désactiver le chiffrement des flux de données

L'`stop-stream-encryption`exemple suivant désactive le chiffrement côté serveur pour le flux spécifié, à l'aide de la clé KMS spécifiée AWS .

```
aws kinesis start-stream-encryption \
    --encryption-type KMS \
    --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-
b334-4d3eb496e452 \
    --stream-name samplestream
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Protection des données dans Amazon Kinesis Data Streams dans le manuel du développeur Amazon Kinesis Data Streams.

- Pour plus de détails sur l'API, reportez-vous StopStreamEncryptionà la section Référence des AWS CLI commandes.

`update-shard-count`

L'exemple de code suivant montre comment utiliser`update-shard-count`.

AWS CLI

Pour mettre à jour le nombre de partitions dans un flux de données

L'`update-shard-count`exemple suivant met à jour le nombre de partitions du flux de données spécifié à 6. Cet exemple utilise une mise à l'échelle uniforme, qui crée des fragments de taille égale.

```
aws kinesis update-shard-count \
    --stream-name samplestream \
    --scaling-type UNIFORM_SCALING \
    --target-shard-count 6
```

Sortie :

```
{
```

```
    "StreamName": "samplestream",
    "CurrentShardCount": 3,
    "TargetShardCount": 6
}
```

Pour plus d'informations, consultez la section Resharding a Stream dans le manuel Amazon Kinesis Data Streams Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateShardCountà la section Référence des AWS CLI commandes.

AWS KMS exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS KMS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-key-deletion`

L'exemple de code suivant montre comment utiliser`cancel-key-deletion`.

AWS CLI

Pour annuler la suppression planifiée d'une clé KMS gérée par le client

L'`cancel-key-deletion`exemple suivant annule la suppression planifiée d'une clé KMS gérée par le client.

```
aws kms cancel-key-deletion \
```

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Lorsque la `cancel-key-deletion` commande aboutit, la suppression planifiée est annulée. Cependant, l'état de la clé KMS est `Disabled` tel que vous ne pouvez pas utiliser la clé KMS dans des opérations cryptographiques. Pour restaurer ses fonctionnalités, utilisez la `enable-key` commande.

Pour plus d'informations, consultez la section Planification et annulation de la suppression de clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous CancelKeyDeletionà la section Référence des AWS CLI commandes.

`connect-custom-key-store`

L'exemple de code suivant montre comment utiliser`connect-custom-key-store`.

AWS CLI

Pour connecter un magasin de clés personnalisé

L'`connect-custom-key-store`exemple suivant reconnecte le magasin de clés personnalisé spécifié. Vous pouvez utiliser une commande comme celle-ci pour connecter un magasin de clés personnalisé pour la première fois ou pour reconnecter un magasin de clés qui a été déconnecté.

Vous pouvez utiliser cette commande pour connecter un magasin de clés AWS CloudHSM ou un magasin de clés externe.

```
aws kms connect-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

Cette commande ne renvoie aucune sortie. Pour vérifier que la commande est efficace, `describe-custom-key-stores` utilisez-la.

Pour plus d'informations sur la connexion d'un magasin de clés AWS CloudHSM, consultez la section Connexion et déconnexion d'un magasin de clés AWS CloudHSM dans le guide du développeur du service de gestion des clés.AWS

Pour plus d'informations sur la connexion d'un magasin de clés externe, voir Connexion et déconnexion d'un magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ConnectCustomKeyStoreà la section Référence des AWS CLI commandes.

`create-alias`

L'exemple de code suivant montre comment utiliser`create-alias`.

AWS CLI

Pour créer un alias pour une clé KMS

La `create-alias` commande suivante crée un alias nommé `example-alias` pour la clé KMS identifiée par son ID de clé`1234abcd-12ab-34cd-56ef-1234567890ab`.

Les noms d'alias doivent commencer par`alias/`. N'utilisez pas de noms d'alias commençant par `alias/aws` ; ils sont réservés à l'usage de AWS.

```
aws kms create-alias \
    --alias-name alias/example-alias \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne renvoie aucune sortie. Pour voir le nouvel alias, utilisez la `list-aliases` commande.

Pour plus d'informations, consultez la section Utilisation d'alias dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous CreateAliasà la section Référence des AWS CLI commandes.

`create-custom-key-store`

L'exemple de code suivant montre comment utiliser`create-custom-key-store`.

AWS CLI

Exemple 1 : pour créer un magasin de AWS clés CloudHSM

L'`create-custom-key-store`exemple suivant crée un magasin de clés AWS CloudHSM soutenu par un cluster AWS CloudHSM à l'aide des paramètres requis. Vous pouvez également ajouter le``custom-key-store-type``parameter with the default value: ``AWS_CLOUDHSM.

Pour spécifier l'entrée de fichier pour la `trust-anchor-certificate` commande dans la AWS CLI, le `file://` préfixe est obligatoire.

```
aws kms create-custom-key-store \
    --custom-key-store-name ExampleCloudHSMKeyStore \
    --cloud-hsm-cluster-id cluster-1a23b4cdefg \
    --key-store-password kmsPswd \
    --trust-anchor-certificate file://customerCA.crt
```

Sortie :

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Pour plus d'informations, consultez la section Création d'un magasin de clés AWS CloudHSM dans AWS le Guide du développeur du service de gestion des clés.

Exemple 2 : pour créer un magasin de clés externe avec une connectivité de point de terminaison public

L'`create-custom-key-store`exemple suivant crée un magasin de clés externe (XKS) qui communique avec AWS KMS via Internet.

Dans cet exemple, le préfixe facultatif est `XksProxyUriPath` utilisé pour. `example-prefix`

REMARQUE : Si vous utilisez la version 1.0 de la AWS CLI, exécutez la commande suivante avant de spécifier un paramètre avec une valeur HTTP ou HTTPS, tel que le XksProxyUriEndpoint paramètre.

```
aws configure set cli_follow_urlparam false
```

Sinon, la version 1.0 de la AWS CLI remplace la valeur du paramètre par le contenu trouvé à cette adresse URI.

```
aws kms create-custom-key-store \
    --custom-key-store-name ExamplePublicEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity PUBLIC_ENDPOINT \
    --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \
    --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Sortie :

```
{
    "CustomKeyStoreId": cks-2234567890abcdef0
}
```

Pour plus d'informations, consultez la section Création d'un magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

Exemple 3 : pour créer un magasin de clés externe avec connectivité au service de point de terminaison VPC

L'`create-custom-key-store`exemple suivant crée un magasin de clés externe (XKS) qui utilise un service de point de terminaison Amazon VPC pour communiquer avec AWS KMS.

REMARQUE : Si vous utilisez la version 1.0 de la AWS CLI, exécutez la commande suivante avant de spécifier un paramètre avec une valeur HTTP ou HTTPS, tel que le XksProxyUriEndpoint paramètre.

```
aws configure set cli_follow_urlparam false
```

Sinon, la version 1.0 de la AWS CLI remplace la valeur du paramètre par le contenu trouvé à cette adresse URI.

```
aws kms create-custom-key-store \
    --custom-key-store-name ExampleVPCEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
    --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \
```

```
    --xks-proxy-uri-path "/kms/xks/v1" \
    --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-example1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Sortie :

```
{
    "CustomKeyStoreId": cks-3234567890abcdef0
}
```

Pour plus d'informations, consultez la section Création d'un magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous CreateCustomKeyStoreà la section Référence des AWS CLI commandes.

`create-grant`

L'exemple de code suivant montre comment utiliser`create-grant`.

AWS CLI

Pour créer une subvention

L'`create-grant`exemple suivant crée une autorisation qui permet à l'`exampleUser`utilisateur d'utiliser la `decrypt` commande sur l'`1234abcd-12ab-34cd-56ef-1234567890ab`exemple de clé KMS. Le `adminRole` rôle est celui du directeur sortant. L'autorisation utilise la `EncryptionContextSubset` contrainte d'autorisation pour autoriser cette autorisation uniquement lorsque le contexte de chiffrement de la `decrypt` demande inclut la paire `"Department": "IT"` clé-valeur.

```
aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::123456789012:user/exampleUser \
    --operations Decrypt \
    --constraints EncryptionContextSubset={Department=IT} \
    --retiring-principal arn:aws:iam::123456789012:role/adminRole
```

Sortie :

```
{
    "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
    "GrantToken": "<grant token here>"
}
```

Pour afficher des informations détaillées sur la subvention, utilisez la `list-grants` commande.

Pour plus d'informations, consultez la section Subventions dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous CreateGrantà la section Référence des AWS CLI commandes.

`create-key`

L'exemple de code suivant montre comment utiliser`create-key`.

AWS CLI

Exemple 1 : pour créer une clé KMS gérée par le client dans AWS KMS

L'`create-key`exemple suivant crée une clé KMS de chiffrement symétrique.

Pour créer la clé KMS de base, une clé de chiffrement symétrique, il n'est pas nécessaire de spécifier de paramètres. Les valeurs par défaut de ces paramètres créent une clé de chiffrement symétrique.

Comme cette commande ne spécifie pas de politique de clé, la clé KMS obtient la politique de clé par défaut pour les clés KMS créées par programmation. Pour afficher la politique clé, utilisez la `get-key-policy` commande. Pour modifier la politique clé, utilisez la `put-key-policy` commande.

```
aws kms create-key
```

La `create-key` commande renvoie les métadonnées clés, y compris l'ID de clé et l'ARN de la nouvelle clé KMS. Vous pouvez utiliser ces valeurs pour identifier la clé KMS dans d'autres opérations AWS KMS. La sortie n'inclut pas les balises. Pour afficher les balises d'une clé KMS, utilisez le`list-resource-tags command`.

Sortie :

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2017-07-05T14:04:55-07:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Remarque : La `create-key` commande ne vous permet pas de spécifier un alias. Pour créer un alias pour la nouvelle clé KMS, utilisez la `create-alias` commande.

Pour plus d'informations, consultez la section Création de clés dans le Guide du développeur du service de gestion des AWS clés.

Exemple 2 : pour créer une clé RSA KMS asymétrique pour le chiffrement et le déchiffrement

L'`create-key`exemple suivant crée une clé KMS contenant une paire de clés RSA asymétrique pour le chiffrement et le déchiffrement. Les spécifications et l'utilisation de la clé ne peuvent pas être modifiées une fois la clé créée. :

```
aws kms create-key \
   --key-spec RSA_4096 \
   --key-usage ENCRYPT_DECRYPT
```

Sortie :

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "RSA_4096",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_4096",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

Pour plus d'informations, consultez la section Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 3 : pour créer une clé KMS à courbe elliptique asymétrique pour la signature et la vérification

Pour créer une clé KMS asymétrique contenant une paire de clés à courbe elliptique asymétrique (ECC) pour la signature et la vérification. Le `--key-usage` paramètre est obligatoire même s'il s'`SIGN_VERIFY`agit de la seule valeur valide pour les clés ECC KMS. Les spécifications et l'utilisation de la clé ne peuvent pas être modifiées une fois la clé créée. :

```
aws kms create-key \
    --key-spec ECC_NIST_P521 \
    --key-usage SIGN_VERIFY
```

Sortie :

```
{
    "KeyMetadata": {
```

```
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ECC_NIST_P521",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

Pour plus d'informations, consultez la section Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 4 : pour créer une clé KMS ML-DSA asymétrique pour la signature et la vérification

Cet exemple crée une clé ML-DSA (Module-Lattice Digital Signature Algorithm) pour la signature et la vérification. Le paramètre d'utilisation des clés est obligatoire même s'il s'`SIGN_VERIFY`agit de la seule valeur valide pour les clés ML-DSA.

```
aws kms create-key \
    --key-spec ML_DSA_65 \
    --key-usage SIGN_VERIFY
```

Sortie :

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "Description": "",
```

```
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ML_DSA_65",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ML_DSA_SHAKE_256"
        ]
    }
}
```

Pour plus d'informations, consultez la section Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 5 : pour créer une clé HMAC KMS

L'`create-key`exemple suivant crée une clé HMAC KMS 384 bits. La `GENERATE_VERIFY_MAC` valeur du `--key-usage` paramètre est obligatoire même s'il s'agit de la seule valeur valide pour les clés HMAC KMS.

```
aws kms create-key \
    --key-spec HMAC_384 \
    --key-usage GENERATE_VERIFY_MAC
```

Sortie :

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "HMAC_384",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "HMAC_384",
        "KeyState": "Enabled",
```

```
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_384"
        ],
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

Pour plus d'informations, consultez la section Clés HMAC dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 6 : pour créer une clé KMS primaire multirégionale

L'`create-key`exemple suivant crée une clé de chiffrement symétrique principale multirégionale. Étant donné que les valeurs par défaut de tous les paramètres créent une clé de chiffrement symétrique, seul le `--multi-region` paramètre est requis pour cette clé KMS. Dans la AWS CLI, pour indiquer qu'un paramètre booléen est vrai, il suffit de spécifier le nom du paramètre.

```
aws kms create-key \
    --multi-region
```

Sortie :

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-09-02T016:15:21-09:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
          "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
```

```
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": []
        },
        "Origin": "AWS_KMS"
    }
}
```

Pour plus d'informations, consultez la section Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 7 : Pour créer une clé KMS pour le matériel clé importé

L'`create-key`exemple suivant crée une clé KMS sans aucun élément clé. Lorsque l'opération est terminée, vous pouvez importer votre propre matériel clé dans la clé KMS. Pour créer cette clé KMS, définissez le `--origin` paramètre sur`EXTERNAL`.

```
aws kms create-key \
    --origin EXTERNAL
```

Sortie :

```
{
     "KeyMetadata": {
         "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "AWSAccountId": "111122223333",
         "CreationDate": "2019-12-02T07:48:55-07:00",
         "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
         "Description": "",
         "Enabled": false,
         "EncryptionAlgorithms": [
             "SYMMETRIC_DEFAULT"
         ],
         "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingImport",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "EXTERNAL"
    }
}
```

Pour plus d'informations, consultez la section Importation de matériel clé dans les clés AWS KMS dans le guide du développeur du service de gestion des AWS clés.

Exemple 6 : pour créer une clé KMS dans un magasin de clés AWS CloudHSM

L'`create-key`exemple suivant crée une clé KMS dans le magasin de clés AWS CloudHSM spécifié. L'opération crée la clé KMS et ses métadonnées dans AWS KMS et crée le matériel clé dans le cluster AWS CloudHSM associé au magasin de clés personnalisé. Les paramètres `--custom-key-store-id` et `--origin` sont obligatoires.

```
aws kms create-key \
    --origin AWS_CLOUDHSM \
    --custom-key-store-id cks-1234567890abcdef0
```

Sortie :

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CloudHsmClusterId": "cluster-1a23b4cdefg",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-1234567890abcdef0",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
```

```
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_CLOUDHSM"
    }
}
```

Pour plus d'informations, consultez les magasins de clés AWS CloudHSM dans AWS le Guide du développeur du service de gestion des clés.

Exemple 8 : pour créer une clé KMS dans un magasin de clés externe

L'`create-key`exemple suivant crée une clé KMS dans le magasin de clés externe spécifié. Les `--xks-key-id` paramètres `--custom-key-store-id--origin`, et sont obligatoires dans cette commande.

Le `--xks-key-id` paramètre indique l'ID d'une clé de chiffrement symétrique existante dans votre gestionnaire de clés externe. Cette clé sert de matériau de clé externe pour la clé KMS. La valeur du `--origin` paramètre doit être `EXTERNAL_KEY_STORE` .Le `custom-key-store-id` paramètre doit identifier un magasin de clés externe connecté à son proxy de magasin de clés externe.

```
aws kms create-key \
    --origin EXTERNAL_KEY_STORE \
    --custom-key-store-id cks-9876543210fedcba9 \
    --xks-key-id bb8562717f809024
```

Sortie :

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-9876543210fedcba9",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
```

```
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "EXTERNAL_KEY_STORE",
        "XksKeyConfiguration": {
            "Id": "bb8562717f809024"
        }
    }
}
```

Pour plus d'informations, consultez la section Stockages de clés externes dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous CreateKeyà la section Référence des AWS CLI commandes.

`decrypt`

L'exemple de code suivant montre comment utiliser`decrypt`.

AWS CLI

Exemple 1 : pour déchiffrer un message chiffré avec une clé KMS symétrique (Linux et macOS)

L'exemple de `decrypt` commande suivant illustre la méthode recommandée pour déchiffrer des données à l'aide de la AWS CLI. Cette version montre comment déchiffrer des données sous une clé KMS symétrique.

Fournissez le texte chiffré dans un fichier. Dans la valeur du `--ciphertext-blob` paramètre, utilisez le `fileb://` préfixe, qui indique à la CLI de lire les données d'un fichier binaire. Si le fichier ne se trouve pas dans le répertoire actuel, saisissez le chemin complet du fichier. Pour plus d'informations sur la lecture des valeurs des paramètres de la AWS CLI depuis un fichier, consultez la section Chargement des paramètres de la AWS CLI depuis un fichier < https:// docs.aws.amazon.com/cli/ latest/userguide/cli - usage-parameters-file .html> dans le guide de l'utilisateur de l'interface de ligne de AWS commande et les meilleures pratiques pour les paramètres de fichiers locaux < https://aws.amazon.com/blogs/ developer/ best-practices-for-local -file-parameters/> dans le blog des outils de ligne de AWS commande. Spécifiez la clé KMS pour déchiffrer le texte chiffré. Le paramètre n'est pas obligatoire lors du déchiffrement avec une

clé KMS symétrique. `--key-id` AWS KMS peut obtenir l'ID de la clé KMS qui a été utilisée pour chiffrer les données à partir des métadonnées du texte chiffré. Toutefois, la spécification de la clé KMS que vous utilisez est une bonne pratique. Cette pratique garantit que vous utilisez la clé KMS comme vous le souhaitez et vous empêche de déchiffrer par inadvertance un texte chiffré à l'aide d'une clé KMS non fiable.Demandez la sortie en texte brut sous forme de valeur de texte.Le paramètre `--query` indique à la CLI de n'obtenir que la valeur du champ à partir de la sortie. `Plaintext` Le `--output` paramètre renvoie la sortie sous forme de texte.Base64 décodez le texte en clair et enregistrez-le dans un fichier. L'exemple suivant montre comment rediriger (|) la valeur du `Plaintext` paramètre vers l'utilitaire Base64, qui le décode. Ensuite, il redirige (>) la sortie décodée vers le `ExamplePlaintext` fichier.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

Cette commande ne produit aucun résultat. La sortie de la `decrypt` commande est décodée en base64 et enregistrée dans un fichier.

Pour plus d'informations, voir Déchiffrer dans le manuel de référence de l'API du service de gestion des AWS clés.

Exemple 2 : pour déchiffrer un message chiffré avec une clé KMS symétrique (invite de commande Windows)

L'exemple suivant est identique au précédent, sauf qu'il utilise l'`certutil`utilitaire pour décoder les données en texte brut en Base64. Cette procédure nécessite deux commandes, comme indiqué dans les exemples suivants.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms decrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
```

```
    --output text ^
    --query Plaintext > ExamplePlaintextFile.base64
```

Exécutez la commande `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Sortie :

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Pour plus d'informations, voir Déchiffrer dans le manuel de référence de l'API du service de gestion des AWS clés.

Exemple 3 : pour déchiffrer un message chiffré avec une clé KMS asymétrique (Linux et macOS)

L'exemple de `decrypt` commande suivant montre comment déchiffrer des données chiffrées sous une clé KMS asymétrique RSA.

Lors de l'utilisation d'une clé KMS asymétrique, le `encryption-algorithm` paramètre, qui spécifie l'algorithme utilisé pour chiffrer le texte en clair, est obligatoire.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

Cette commande ne produit aucun résultat. La sortie de la `decrypt` commande est décodée en base64 et enregistrée dans un fichier.

Pour plus d'informations, consultez la section Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, voir Déchiffrer dans la référence des AWS CLI commandes.

`delete-alias`

L'exemple de code suivant montre comment utiliser`delete-alias`.

AWS CLI

Pour supprimer un alias AWS KMS

L'`delete-alias`exemple suivant supprime l'alias`alias/example-alias`. Le nom de l'alias doit commencer par alias/.

```
aws kms delete-alias \
    --alias-name alias/example-alias
```

Cette commande ne produit aucun résultat. Pour trouver l'alias, utilisez la `list-aliases` commande.

Pour plus d'informations, consultez la section Suppression d'un alias dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DeleteAliasà la section Référence des AWS CLI commandes.

`delete-custom-key-store`

L'exemple de code suivant montre comment utiliser`delete-custom-key-store`.

AWS CLI

Pour supprimer un magasin de clés personnalisé

L'`delete-custom-key-store`exemple suivant supprime le magasin de clés personnalisé spécifié.

La suppression d'un magasin de clés AWS CloudHSM n'a aucun effet sur le cluster CloudHSM associé. La suppression d'une banque de clés externe n'a aucun effet sur le proxy de banque de clés externe, le gestionnaire de clés externe ou les clés externes associés.

REMARQUE : Avant de supprimer un magasin de clés personnalisé, vous devez planifier la suppression de toutes les clés KMS du magasin de clés personnalisé, puis attendre que ces clés KMS soient supprimées. Ensuite, vous devez déconnecter le magasin de clés personnalisé. Pour

obtenir de l'aide pour trouver les clés KMS dans votre magasin de clés personnalisé, consultez la section Supprimer un magasin de clés AWS CloudHSM (API) dans AWS le guide du développeur du service de gestion des clés.

```
delete-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

Cette commande ne renvoie aucune sortie. Pour vérifier que le magasin de clés personnalisé est supprimé, utilisez la `describe-custom-key-stores` commande.

Pour plus d'informations sur la suppression d'une AWS banque de clés CloudHSM, consultez la section Suppression d'une banque de clés AWS CloudHSM dans le guide du développeur du service de gestion des clés.AWS

Pour plus d'informations sur la suppression de banques de clés externes, consultez la section Suppression d'une banque de clés externe dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DeleteCustomKeyStoreà la section Référence des AWS CLI commandes.

`delete-imported-key-material`

L'exemple de code suivant montre comment utiliser`delete-imported-key-material`.

AWS CLI

Pour supprimer le contenu clé importé d'une clé KMS

L'`delete-imported-key-material`exemple suivant supprime le matériel clé qui a été importé dans une clé KMS.

```
aws kms delete-imported-key-material \
   --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
```

```
}
```

Pour plus d'informations, consultez la section Suppression des éléments clés importés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DeleteImportedKeyMaterialà la section Référence des AWS CLI commandes.

`derive-shared-secret`

L'exemple de code suivant montre comment utiliser`derive-shared-secret`.

AWS CLI

Pour obtenir un secret partagé

L'`derive-shared-secret`exemple suivant déduit un secret partagé à l'aide d'un algorithme d'accord de clés.

Vous devez utiliser une paire de clés KMS asymétrique recommandée par le NIST (ECC) ou (régions de SM2 Chine uniquement) avec une `KeyUsage` valeur de to call. `KEY_AGREEMENT` DeriveSharedSecret

```
aws kms derive-shared-secret \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-agreement-algorithm ECDH \
    --public-
key "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvH3Yj0wbkLEpUl95Cv1cJVjsVNSjwGq3tCLnzXfhVwV
```

Sortie :

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "SharedSecret": "MEYCIQCKZLWyTk5runarx6XiAkU9gv3lbwPO/pHa
+DXFehzdDwIhANwpsIV2g/9SPWLLsF6p/hiSskuIXMTRwqrMdVKWTMHG",
    "KeyAgreementAlgorithm": "ECDH",
    "KeyOrigin": "AWS_KMS"
}
```

Pour plus d'informations, consultez la référence DeriveSharedSecretde l'API du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DeriveSharedSecretà la section Référence des AWS CLI commandes.

`describe-custom-key-stores`

L'exemple de code suivant montre comment utiliser`describe-custom-key-stores`.

AWS CLI

Exemple 1 : pour obtenir des informations sur un magasin de AWS clés CloudHSM

L'`describe-custom-key-store`exemple suivant affiche des détails sur le magasin de clés AWS CloudHSM spécifié. La commande est la même pour tous les types de magasins de clés personnalisés, mais le résultat diffère selon le type de magasin de clés et, dans le cas d'un magasin de clés externe, son option de connectivité.

Par défaut, cette commande affiche des informations sur tous les magasins de clés personnalisés du compte et de la région. Pour afficher des informations sur un magasin de clés personnalisé en particulier, utilisez le `custom-key-store-id` paramètre `custom-key-store-name` or.

```
aws kms describe-custom-key-stores \
    --custom-key-store-name ExampleCloudHSMKeyStore
```

Le résultat de cette commande inclut des informations utiles sur le magasin de clés AWS CloudHSM, notamment son état de connexion (). `ConnectionState` Si l'état de connexion est `FAILED` défini comme tel, la sortie inclut un `ConnectionErrorCode` champ qui décrit le problème.

Sortie :

```
{
    "CustomKeyStores": [
        {
            "CloudHsmClusterId": "cluster-1a23b4cdefg",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-04-05T14:04:55-07:00",
            "CustomKeyStoreId": "cks-1234567890abcdef0",
            "CustomKeyStoreName": "ExampleExternalKeyStore",
            "TrustAnchorCertificate": "<certificate appears here>"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Affichage d'un magasin de clés AWS CloudHSM dans AWS le Guide du développeur du service de gestion des clés.

Exemple 2 : pour obtenir des informations sur un magasin de clés externe connecté à un point de terminaison public

L'`describe-custom-key-store`exemple suivant affiche des détails sur le magasin de clés externe spécifié. La commande est la même pour tous les types de magasins de clés personnalisés, mais le résultat diffère selon le type de magasin de clés et, dans le cas d'un magasin de clés externe, son option de connectivité.

Par défaut, cette commande affiche des informations sur tous les magasins de clés personnalisés du compte et de la région. Pour afficher des informations sur un magasin de clés personnalisé en particulier, utilisez le `custom-key-store-id` paramètre `custom-key-store-name` or.

```
aws kms describe-custom-key-stores \
    --custom-key-store-id cks-9876543210fedcba9
```

Le résultat de cette commande inclut des détails utiles sur le magasin de clés externe, notamment son état de connexion (`ConnectionState`). Si l'état de connexion est `FAILED` défini comme tel, la sortie inclut un `ConnectionErrorCode` champ qui décrit le problème.

Sortie :

```
{
    "CustomKeyStores": [
        {
            "CustomKeyStoreId": "cks-9876543210fedcba9",
            "CustomKeyStoreName": "ExampleXKS",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-12-02T07:48:55-07:00",
            "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
            "XksProxyConfiguration": {
                "AccessKeyId": "ABCDE12345670EXAMPLE",
                "Connectivity": "PUBLIC_ENDPOINT",
                "UriEndpoint": "https://myproxy.xks.example.com",
                "UriPath": "/example-prefix/kms/xks/v1"
            }
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'un magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

Exemple 3 : pour obtenir des informations sur un magasin de clés externe doté d'une connectivité au service de point de terminaison VPC

L'`describe-custom-key-store`exemple suivant affiche des détails sur le magasin de clés externe spécifié. La commande est la même pour tous les types de magasins de clés personnalisés, mais le résultat diffère selon le type de magasin de clés et, dans le cas d'un magasin de clés externe, son option de connectivité.

Par défaut, cette commande affiche des informations sur tous les magasins de clés personnalisés du compte et de la région. Pour afficher des informations sur un magasin de clés personnalisé en particulier, utilisez le `custom-key-store-id` paramètre `custom-key-store-name` or.

```
aws kms describe-custom-key-stores \
    --custom-key-store-id cks-2234567890abcdef0
```

Le résultat de cette commande inclut des détails utiles sur le magasin de clés externe, notamment son état de connexion (`ConnectionState`). Si l'état de connexion est `FAILED` défini comme tel, la sortie inclut un `ConnectionErrorCode` champ qui décrit le problème.

Sortie :

```
{
    "CustomKeyStores": [
        {
            "CustomKeyStoreId": "cks-3234567890abcdef0",
            "CustomKeyStoreName": "ExampleVPCExternalKeyStore",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-12-22T07:48:55-07:00",
            "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
            "XksProxyConfiguration": {
                "AccessKeyId": "ABCDE12345670EXAMPLE",
                "Connectivity": "VPC_ENDPOINT_SERVICE",
                "UriEndpoint": "https://myproxy-private.xks.example.com",
                "UriPath": "/kms/xks/v1",
                "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-
example1"
```

```
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'un magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DescribeCustomKeyStoresà la section Référence des AWS CLI commandes.

`describe-key`

L'exemple de code suivant montre comment utiliser`describe-key`.

AWS CLI

Exemple 1 : pour trouver des informations détaillées sur une clé KMS

L'`describe-key`exemple suivant fournit des informations détaillées sur la clé AWS gérée pour Amazon S3 dans l'exemple de compte et de région. Vous pouvez utiliser cette commande pour obtenir des informations sur les clés AWS gérées et les clés gérées par le client.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise une valeur de nom d'alias, mais vous pouvez utiliser un ID de clé, un ARN de clé, un nom d'alias ou un ARN d'alias dans cette commande.

```
aws kms describe-key \
    --key-id alias/aws/s3
```

Sortie :

```
{
    "KeyMetadata": {
        "AWSAccountId": "846764612917",
        "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
        "Arn": "arn:aws:kms:us-west-2:846764612917:key/
b8a9477d-836c-491f-857e-07937918959b",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CreationDate": 2017-06-30T21:44:32.140000+00:00,
        "Enabled": true,
```

```
        "Description": "Default KMS key that protects my S3 objects when no other
 key is defined",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "AWS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Pour plus d'informations, consultez la section Affichage des clés dans le Guide du développeur du service de gestion des AWS clés.

Exemple 2 : pour obtenir des informations sur une clé KMS asymétrique RSA

L'`describe-key`exemple suivant fournit des informations détaillées sur une clé RSA KMS asymétrique utilisée pour la signature et la vérification.

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2019-12-02T19:47:14.861000+00:00",
        "CustomerMasterKeySpec": "RSA_2048",
        "Enabled": false,
        "Description": "",
        "KeyState": "Disabled",
        "Origin": "AWS_KMS",
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_2048",
        "KeyUsage": "SIGN_VERIFY",
        "SigningAlgorithms": [
```

```
            "RSASSA_PKCS1_V1_5_SHA_256",
            "RSASSA_PKCS1_V1_5_SHA_384",
            "RSASSA_PKCS1_V1_5_SHA_512",
            "RSASSA_PSS_SHA_256",
            "RSASSA_PSS_SHA_384",
            "RSASSA_PSS_SHA_512"
        ]
    }
}
```

Exemple 3 : pour obtenir des informations sur une clé de réplique multirégionale

L'`describe-key`exemple suivant permet d'obtenir les métadonnées d'une clé de réplique multirégionale. Cette clé multirégionale est une clé de chiffrement symétrique. La sortie d'une `describe-key` commande pour une clé multirégionale renvoie des informations sur la clé primaire et toutes ses répliques.

```
aws kms describe-key \
    --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Sortie :

```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": "2021-06-28T21:09:16.114000+00:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
```

```
        ],
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-northeast-1"
                },
                {
                    "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "sa-east-1"
                }
            ]
        }
    }
}
```

Exemple 4 : pour obtenir des informations sur une clé HMAC KMS

L'`describe-key`exemple suivant permet d'obtenir des informations détaillées sur une clé HMAC KMS.

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyMetadata": {
        "AWSAccountId": "123456789012",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
        "Arn": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2022-04-03T22:23:10.194000+00:00",
        "Enabled": true,
        "Description": "Test key",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "HMAC_256",
        "MacAlgorithms": [
            "HMAC_SHA_256"
        ],
        "MultiRegion": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeKeyà la section Référence des AWS CLI commandes.

`disable-key-rotation`

L'exemple de code suivant montre comment utiliser`disable-key-rotation`.

AWS CLI

Pour désactiver la rotation automatique d'une clé KMS

L'`disable-key-rotation`exemple suivant désactive la rotation automatique d'une clé KMS gérée par le client. Pour réactiver la rotation automatique, utilisez la `enable-key-rotation` commande.

```
aws kms disable-key-rotation \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne produit aucun résultat. Pour vérifier que la rotation automatique est désactivée pour la clé KMS, utilisez la `get-key-rotation-status` commande.

Pour plus d'informations, voir Rotation des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DisableKeyRotationà la section Référence des AWS CLI commandes.

`disable-key`

L'exemple de code suivant montre comment utiliser`disable-key`.

AWS CLI

Pour désactiver temporairement une clé KMS

La `disable-key` commande suivante désactive une clé KMS gérée par le client. Pour réactiver la clé KMS, utilisez la `enable-key` commande.

```
aws kms disable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation et désactivation des clés dans le guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DisableKeyà la section Référence des AWS CLI commandes.

`disconnect-custom-key-store`

L'exemple de code suivant montre comment utiliser`disconnect-custom-key-store`.

AWS CLI

Pour déconnecter un magasin de clés personnalisé

L'`disconnect-custom-key-store`exemple suivant déconnecte un magasin de clés personnalisé de son cluster AWS CloudHSM. Vous pouvez déconnecter un magasin de clés pour résoudre un problème, pour mettre à jour ses paramètres ou pour empêcher l'utilisation des clés KMS du magasin de clés dans des opérations cryptographiques.

Cette commande est identique pour tous les magasins de clés personnalisés, y compris les magasins de clés AWS CloudHSM et les magasins de clés externes.

Avant d'exécuter la commande, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

```
$ aws kms disconnect-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

Cette commande ne produit aucune sortie. Vérifiez que la commande est efficace, utilisez-la. `describe-custom-key-stores`

Pour plus d'informations sur la déconnexion d'un magasin de clés AWS CloudHSM, consultez la section Connexion et déconnexion d'un magasin de clés AWS CloudHSM dans le guide du développeur du service de gestion des clés.AWS

Pour plus d'informations sur la déconnexion d'un magasin de clés externe, voir Connexion et déconnexion d'un magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous DisconnectCustomKeyStoreà la section Référence des AWS CLI commandes.

`enable-key-rotation`

L'exemple de code suivant montre comment utiliser`enable-key-rotation`.

AWS CLI

Pour activer la rotation automatique d'une clé KMS

L'`enable-key-rotation`exemple suivant permet la rotation automatique d'une clé KMS gérée par le client avec une période de rotation de 180 jours. La clé KMS fera l'objet d'une rotation d'un an (environ 365 jours) à compter de la date d'exécution de cette commande et chaque année par la suite.

Le `--key-id` paramètre identifie la clé KMS. Cet exemple utilise une valeur ARN clé, mais vous pouvez utiliser l'ID de clé ou l'ARN de la clé KMS. Le `--rotation-period-in-days` paramètre indique le nombre de jours entre chaque date de rotation. Spécifiez une valeur comprise entre 90 et 2560 jours. Si aucune valeur n'est spécifiée, la valeur par défaut est de 365 jours.

```
aws kms enable-key-rotation \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --rotation-period-in-days 180
```

Cette commande ne produit aucun résultat. Pour vérifier que la clé KMS est activée, utilisez la `get-key-rotation-status` commande.

Pour plus d'informations, voir Rotation des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous EnableKeyRotationà la section Référence des AWS CLI commandes.

`enable-key`

L'exemple de code suivant montre comment utiliser`enable-key`.

AWS CLI

Pour activer une clé KMS

L'`enable-key`exemple suivant active une clé gérée par le client. Vous pouvez utiliser une commande comme celle-ci pour activer une clé KMS que vous avez temporairement désactivée à l'aide de la `disable-key` commande. Vous pouvez également l'utiliser pour activer une clé KMS qui est désactivée car sa suppression a été planifiée et la suppression a été annulée.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise une valeur d'ID de clé, mais vous pouvez utiliser une valeur d'ID de clé ou une valeur d'ARN de clé dans cette commande.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un identifiant valide.

```
aws kms enable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne produit aucun résultat. Pour vérifier que la clé KMS est activée, utilisez la `describe-key` commande. Consultez les valeurs des `Enabled` champs `KeyState` et dans la `describe-key` sortie.

Pour plus d'informations, consultez la section Activation et désactivation des clés dans le guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous EnableKeyà la section Référence des AWS CLI commandes.

`encrypt`

L'exemple de code suivant montre comment utiliser`encrypt`.

AWS CLI

Exemple 1 : chiffrer le contenu d'un fichier sous Linux ou macOS

La `encrypt` commande suivante montre la méthode recommandée pour chiffrer les données avec la AWS CLI.

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

La commande effectue plusieurs opérations :

Utilise le `--plaintext` paramètre pour indiquer les données à chiffrer. La valeur de ce paramètre doit être codée en base64. La valeur du `plaintext` paramètre doit être codée en base64, ou vous devez utiliser le préfixe`fileb://`, qui indique à la AWS CLI de lire les données binaires du fichier. Si le fichier ne se trouve pas dans le répertoire actuel, tapez le chemin complet du fichier. Par exemple, `fileb:///var/tmp/ExamplePlaintextFile` ou `fileb://C:\Temp\ExamplePlaintextFile`. Pour plus d'informations sur la lecture des valeurs des paramètres de la AWS CLI depuis un fichier, consultez la section AWS Chargement de paramètres depuis un fichierdans le guide de l'utilisateur de l'interface de ligne de AWS commande et les meilleures pratiques pour les paramètres de fichiers locaux sur le `--output` blog des outils de ligne de commande. Ces `--query` paramètres extraient les données chiffrées, appelées texte chiffré, de la sortie de la commande.Pour plus d'informations sur le contrôle de la sortie, voir Contrôle de la commande Sortie dans le guide de l'utilisateur de l'interface de ligne de AWS commande. Utilise l'`base64`utilitaire pour décoder la sortie extraite en données binaires. Le texte chiffré renvoyé par une commande réussie `encrypt` est du texte codé en base64. Vous devez décoder ce texte avant de pouvoir utiliser la AWS CLI pour le déchiffrer. Enregistre le texte chiffré binaire dans un fichier.La dernière partie de la commande (`> ExampleEncryptedFile`) enregistre le texte chiffré binaire dans un fichier pour faciliter le déchiffrement. Pour un exemple de commande utilisant la AWS CLI pour déchiffrer des données, consultez les exemples de déchiffrement.

Exemple 2 : utilisation de la AWS CLI pour chiffrer des données sous Windows

Cet exemple est le même que le précédent, sauf qu'il utilise l'`certutil`outil au lieu de`base64`. Cette procédure nécessite deux commandes, comme indiqué dans l'exemple suivant.

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64

certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

Exemple 3 : Chiffrement avec une clé KMS asymétrique

La `encrypt` commande suivante montre comment chiffrer du texte en clair avec une clé KMS asymétrique. Le paramètre `--encryption-algorithm` est obligatoire. Comme dans toutes les commandes de la `encrypt` CLI, le `plaintext` paramètre doit être codé en base64, ou vous devez utiliser le `fileb://` préfixe, qui indique à la AWS CLI de lire les données binaires du fichier.

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, voir Encrypt in AWS CLI Command Reference.

`generate-data-key-pair-without-plaintext`

L'exemple de code suivant montre comment utiliser`generate-data-key-pair-without-plaintext`.

AWS CLI

Pour générer une paire de clés de données asymétriques ECC NIST P384

L'`generate-data-key-pair-without-plaintext`exemple suivant demande une paire de clés ECC NIST P384 pour une utilisation en dehors de. AWS

La commande renvoie une clé publique en texte brut et une copie de la clé privée chiffrée sous la clé KMS spécifiée. Il ne renvoie pas de clé privée en texte brut. Vous pouvez stocker en toute sécurité la clé privée chiffrée avec les données chiffrées et appeler AWS KMS pour déchiffrer la clé privée lorsque vous devez l'utiliser.

Pour demander une paire de clés de données asymétriques ECC NIST P384, utilisez le `key-pair-spec` paramètre avec une valeur de. `ECC_NIST_P384`

La clé KMS que vous spécifiez doit être une clé KMS de chiffrement symétrique, c'est-à-dire une clé KMS dont `KeySpec` la valeur est de`SYMMETRIC_DEFAULT`.

REMARQUE : Les valeurs de la sortie de cet exemple sont tronquées pour être affichées.

```
aws kms generate-data-key-pair-without-plaintext \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-pair-spec ECC_NIST_P384
```

Sortie :

```
{
    "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
    "PublicKey":
 "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbdOyND
+NmYDH/mL1OSIEuLrcdZ5hrMH4pk83r40l...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "KeyPairSpec": "ECC_NIST_P384"
}
```

Les `PublicKey` et `PrivateKeyCiphertextBlob` sont renvoyés au format codé en base64.

Pour plus d'informations, consultez la section Paires de clés de données dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GenerateDataKeyPairWithoutPlaintextà la section Référence des AWS CLI commandes.

`generate-data-key-pair`

L'exemple de code suivant montre comment utiliser`generate-data-key-pair`.

AWS CLI

Pour générer une paire de clés de données asymétriques RSA de 2048 bits

L'`generate-data-key-pair`exemple suivant demande une paire de clés de données asymétriques RSA de 2048 bits pour une utilisation en dehors de. AWS La commande renvoie une clé publique en clair et une clé privée en texte clair pour une utilisation et une suppression immédiates, ainsi qu'une copie de la clé privée chiffrée sous la clé KMS spécifiée. Vous pouvez stocker en toute sécurité la clé privée cryptée avec les données cryptées.

Pour demander une paire de clés de données asymétriques RSA de 2048 bits, utilisez le `key-pair-spec` paramètre avec une valeur de. `RSA_2048`

La clé KMS que vous spécifiez doit être une clé KMS de chiffrement symétrique, c'est-à-dire une clé KMS dont `KeySpec` la valeur est de`SYMMETRIC_DEFAULT`.

REMARQUE : Les valeurs de la sortie de cet exemple sont tronquées pour être affichées.

```
aws kms generate-data-key-pair \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-pair-spec RSA_2048
```

Sortie :

```
{
    "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
    "PrivateKeyPlaintext": "MIIG/
QIBADANBgkqhkiG9w0BAQEFAASCBucwggbjAgEAAoIBgQDcDd4YzI
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P42ZgMf+YvU5IgS4ut...",
    "PublicKey":
 "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbdOyND
+NmYDH/mL1OSIEuLrcdZ5hrMH4pk83r40l...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
    "KeyPairSpec": "RSA_2048"
```

```
}
```

Les `PublicKey`PrivateKeyPlaintext, et `PrivateKeyCiphertextBlob` sont renvoyés au format codé en base64.

Pour plus d'informations, consultez la section Paires de clés de données dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GenerateDataKeyPairà la section Référence des AWS CLI commandes.

`generate-data-key-without-plaintext`

L'exemple de code suivant montre comment utiliser`generate-data-key-without-plaintext`.

AWS CLI

Pour générer une clé de données symétrique de 256 bits sans clé en texte brut

L'`generate-data-key-without-plaintext`exemple suivant demande une copie cryptée d'une clé de données symétrique de 256 bits pour une utilisation en dehors de. AWS Vous pouvez appeler AWS KMS pour déchiffrer la clé de données lorsque vous êtes prêt à l'utiliser.

Pour demander une clé de données de 256 bits, utilisez le `key-spec` paramètre avec une valeur de. `AES_256` Pour demander une clé de données de 128 bits, utilisez le `key-spec` paramètre avec une valeur de. `AES_128` Pour toutes les autres longueurs de clé de données, utilisez le `number-of-bytes` paramètre.

La clé KMS que vous spécifiez doit être une clé KMS de chiffrement symétrique, c'est-à-dire une clé KMS dont la valeur de spécification de clé est SYMMETRIC_DEFAULT.

```
aws kms generate-data-key-without-plaintext \
    --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --key-spec AES_256
```

Sortie :

```
{
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
```

```
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

La `CiphertextBlob` (clé de données cryptée) est renvoyée au format codé en base64.

Pour plus d'informations, consultez la section Clés de données dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GenerateDataKeyWithoutPlaintextà la section Référence des AWS CLI commandes.

`generate-data-key`

L'exemple de code suivant montre comment utiliser`generate-data-key`.

AWS CLI

Exemple 1 : pour générer une clé de données symétrique de 256 bits

L'`generate-data-key`exemple suivant demande une clé de données symétrique de 256 bits à utiliser en dehors de. AWS La commande renvoie une clé de données en texte brut pour une utilisation et une suppression immédiates, ainsi qu'une copie de cette clé de données chiffrée sous la clé KMS spécifiée. Vous pouvez stocker en toute sécurité la clé de données chiffrée avec les données chiffrées.

Pour demander une clé de données de 256 bits, utilisez le `key-spec` paramètre avec une valeur de. `AES_256` Pour demander une clé de données de 128 bits, utilisez le `key-spec` paramètre avec une valeur de. `AES_128` Pour toutes les autres longueurs de clé de données, utilisez le `number-of-bytes` paramètre.

La clé KMS que vous spécifiez doit être une clé KMS de chiffrement symétrique, c'est-à-dire une clé KMS dont la valeur de spécification de clé est SYMMETRIC_DEFAULT.

```
aws kms generate-data-key \
    --key-id alias/ExampleAlias \
    --key-spec AES_256
```

Sortie :

```
{
    "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

La `Plaintext` (clé de données en texte brut) et la `CiphertextBlob` (clé de données cryptée) sont renvoyées au format codé en base64.

Pour plus d'informations, consultez la section Clés de données dans le Guide du développeur du service de gestion des AWS clés. Exemple 2 : pour générer une clé de données symétrique 512 bits

L'`generate-data-key`exemple suivant demande une clé de données symétrique de 512 bits pour le chiffrement et le déchiffrement. La commande renvoie une clé de données en texte brut pour une utilisation et une suppression immédiates, ainsi qu'une copie de cette clé de données chiffrée sous la clé KMS spécifiée. Vous pouvez stocker en toute sécurité la clé de données chiffrée avec les données chiffrées.

Pour demander une longueur de clé autre que 128 ou 256 bits, utilisez le `number-of-bytes` paramètre. Pour demander une clé de données 512 bits, l'exemple suivant utilise le `number-of-bytes` paramètre avec une valeur de 64 (octets).

La clé KMS que vous spécifiez doit être une clé KMS de chiffrement symétrique, c'est-à-dire une clé KMS dont la valeur de spécification de clé est SYMMETRIC_DEFAULT.

REMARQUE : Les valeurs de la sortie de cet exemple sont tronquées pour être affichées.

```
aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --number-of-bytes 64
```

Sortie :

```
{
```

```
    "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCAlUwggJRAgEAMIICSgYJKoZ...",
    "Plaintext": "ty8Lr0Bk6OF07M2BWt6qbFdNB
+G00ZLtf5MSEb4al3R2UKWGOp06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm4OovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

Les `Plaintext` (clé de données en texte brut) et `CiphertextBlob` (clé de données cryptée) sont renvoyées au format codé en base64.

Pour plus d'informations, consultez la section Clés de données dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GenerateDataKeyà la section Référence des AWS CLI commandes.

`generate-mac`

L'exemple de code suivant montre comment utiliser`generate-mac`.

AWS CLI

Exemple 1 : Pour générer un HMAC pour un message

La `generate-mac` commande suivante génère un HMAC pour un message, une clé HMAC KMS et un algorithme MAC. L'algorithme doit être pris en charge par la clé HMAC KMS spécifiée.

Dans la AWS CLI v2, la valeur du `message` paramètre doit être codée en Base64. Vous pouvez également enregistrer le message dans un fichier et utiliser le `fileb://` préfixe, qui indique à la AWS CLI de lire les données binaires du fichier.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte. L'ID de clé doit représenter une clé HMAC KMS dont l'utilisation de `GENERATE_VERIFY_MAC` la clé est.

```
msg=(echo 'Hello World' | base64)

aws kms generate-mac \
```

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://Message \
    --mac-algorithm HMAC_SHA_384
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "Mac": "<HMAC_TAG>",
    "MacAlgorithm": "HMAC_SHA_384"
}
```

Pour plus d'informations sur l'utilisation des clés HMAC KMS dans AWS KMS, consultez la section Clés HMAC dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 2 : Pour enregistrer un HMAC dans un fichier (Linux et macOS)

L'`generate-mac`exemple suivant génère un HMAC pour un message court stocké dans un fichier local. La commande obtient également la `Mac` propriété de la réponse, la décode en Base64 et l'enregistre dans le fichier. ExampleMac Vous pouvez utiliser le fichier MAC dans une `verify-mac` commande qui vérifie le MAC.

La `generate-mac` commande nécessite un message codé en Base64 et un algorithme MAC pris en charge par votre clé HMAC KMS. Pour obtenir les algorithmes MAC pris en charge par votre clé KMS, utilisez la `describe-key` commande.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte. L'ID de clé doit représenter une clé KMS asymétrique avec une utilisation de la clé GENERATE_VERIFY_MAC.

```
echo 'hello world' | base64 > EncodedMessage

aws kms generate-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --mac-algorithm HMAC_SHA_384 \
    --output text \
    --query Mac | base64 --decode > ExampleMac
```

Cette commande ne produit aucun résultat. Cet exemple extrait la `Mac` propriété de la sortie et l'enregistre dans un fichier.

Pour plus d'informations sur l'utilisation des clés HMAC KMS dans AWS KMS, consultez la section Clés HMAC dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GenerateMacà la section Référence des AWS CLI commandes.

`generate-random`

L'exemple de code suivant montre comment utiliser`generate-random`.

AWS CLI

Exemple 1 : pour générer une chaîne d'octets aléatoire de 256 bits (Linux ou macOS)

L'`generate-random`exemple suivant génère une chaîne d'octets aléatoire de 256 bits (32 octets) codée en base64. L'exemple décode la chaîne d'octets et l'enregistre dans le fichier aléatoire.

Lorsque vous exécutez cette commande, vous devez utiliser le `number-of-bytes` paramètre pour spécifier la longueur de la valeur aléatoire en octets.

Vous ne spécifiez pas de clé KMS lorsque vous exécutez cette commande. La chaîne d'octets aléatoire n'est liée à aucune clé KMS.

Par défaut, AWS KMS génère le nombre aléatoire. Toutefois, si vous spécifiez un magasin de clés personnalisé, la chaîne d'octets aléatoire est générée dans le cluster AWS CloudHSM associé au magasin de clés personnalisé.

Cet exemple utilise les paramètres et valeurs suivants :

Il utilise le `--number-of-bytes` paramètre requis avec une valeur de `32` pour demander une chaîne de 32 octets (256 bits). Il utilise le `--output` paramètre avec une valeur de pour demander à la AWS CLI de renvoyer la sortie sous forme de texte, au lieu de JSON. Il utilise le `--query parameter` pour extraire la valeur de la `Plaintext` propriété de la réponse. Il envoie (|) la sortie de la commande à l'`base64`utilitaire, qui décode la sortie extraite. Il utilise l'opérateur de `text` redirection (**>**) pour enregistrer la chaîne d'octets décodée dans le fichier .it

utilise l'opérateur de redirection (>) `ExampleRandom` pour enregistrer le texte chiffré binaire dans un fichier.

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext | base64 --decode > ExampleRandom
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la référence GenerateRandomde l'API du service de gestion des AWS clés.

Exemple 2 : pour générer un nombre aléatoire de 256 bits (invite de commande Windows)

L'exemple suivant utilise la `generate-random` commande pour générer une chaîne d'octets aléatoire de 256 bits (32 octets) codée en base64. L'exemple décode la chaîne d'octets et l'enregistre dans le fichier aléatoire. Cet exemple est identique à l'exemple précédent, sauf qu'il utilise l'`certutil`utilitaire de Windows pour décoder en base64 la chaîne d'octets aléatoire avant de l'enregistrer dans un fichier.

Tout d'abord, générez une chaîne d'octets aléatoire codée en base64 et enregistrez-la dans un fichier temporaire. `ExampleRandom.base64`

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext > ExampleRandom.base64
```

La sortie de la `generate-random` commande étant enregistrée dans un fichier, cet exemple ne produit aucune sortie.

Utilisez maintenant la `certutil -decode` commande pour décoder la chaîne d'octets codée en base64 dans le fichier. `ExampleRandom.base64` Ensuite, il enregistre la chaîne d'octets décodée dans le `ExampleRandom` fichier.

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Sortie :

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Pour plus d'informations, consultez la référence GenerateRandomde l'API du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GenerateRandomà la section Référence des AWS CLI commandes.

`get-key-policy`

L'exemple de code suivant montre comment utiliser`get-key-policy`.

AWS CLI

Pour copier une politique clé d'une clé KMS vers une autre clé KMS

L'`get-key-policy`exemple suivant extrait la politique clé d'une clé KMS et l'enregistre dans un fichier texte. Il remplace ensuite la politique d'une autre clé KMS en utilisant le fichier texte comme entrée de politique.

Comme le `--policy` paramètre de `put-key-policy` nécessite une chaîne, vous devez utiliser l'`--output text`option pour renvoyer la sortie sous forme de chaîne de texte au lieu de JSON.

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --query Policy \
    --output text > policy.txt

aws kms put-key-policy \
    --policy-name default \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --policy file://policy.txt
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez le PutKeyPolicymanuel de référence de l'API AWS KMS.

- Pour plus de détails sur l'API, reportez-vous GetKeyPolicyà la section Référence des AWS CLI commandes.

`get-key-rotation-status`

L'exemple de code suivant montre comment utiliser`get-key-rotation-status`.

AWS CLI

Pour récupérer l'état de rotation d'une clé KMS.

L'`get-key-rotation-status`exemple suivant renvoie des informations sur l'état de rotation de la clé KMS spécifiée, notamment si la rotation automatique est activée, la période de rotation et la prochaine date de rotation planifiée. Vous pouvez utiliser cette commande sur les clés KMS gérées par le client et les clés KMS AWS gérées. Cependant, toutes les clés KMS AWS gérées font l'objet d'une rotation automatique chaque année.

```
aws kms get-key-rotation-status \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": true,
    "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",
    "RotationPeriodInDays": 365
}
```

Pour plus d'informations, voir Rotation des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GetKeyRotationStatusà la section Référence des AWS CLI commandes.

`get-parameters-for-import`

L'exemple de code suivant montre comment utiliser`get-parameters-for-import`.

AWS CLI

Pour obtenir les éléments nécessaires à l'importation de matériel clé dans une clé KMS

L'`get-parameters-for-import`exemple suivant permet d'obtenir la clé publique et le jeton d'importation dont vous avez besoin pour importer le matériel clé dans une clé KMS. Lorsque

vous utilisez la `import-key-material` commande, veillez à utiliser le jeton d'importation et le contenu clé chiffrés par la clé publique renvoyés dans la même `get-parameters-for-import` commande. En outre, l'algorithme d'encapsulage que vous spécifiez dans cette commande doit être celui que vous utilisez pour chiffrer le contenu clé avec la clé publique.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise un ID de clé, mais vous pouvez utiliser un ID de clé ou un ARN de clé dans cette commande.

```
aws kms get-parameters-for-import \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --wrapping-algorithm RSAES_OAEP_SHA_256 \
    --wrapping-key-spec RSA_2048
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "PublicKey": "<public key base64 encoded data>",
    "ImportToken": "<import token base64 encoded data>",
    "ParametersValidTo": 1593893322.32
}
```

Pour plus d'informations, voir Télécharger la clé publique et le jeton d'importation dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GetParametersForImportà la section Référence des AWS CLI commandes.

`get-public-key`

L'exemple de code suivant montre comment utiliser`get-public-key`.

AWS CLI

Exemple 1 : pour télécharger la clé publique d'une clé KMS asymétrique

L'`get-public-key`exemple suivant télécharge la clé publique d'une clé KMS asymétrique.

Outre le renvoi de la clé publique, le résultat inclut les informations dont vous avez besoin pour utiliser la clé publique en toute sécurité en dehors de AWS KMS, notamment l'utilisation de la clé et les algorithmes de chiffrement pris en charge.

```
aws kms get-public-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "PublicKey": "jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAl5epvg1/
QtJhxSi2g9SDEVg8QV/...",
    "CustomerMasterKeySpec": "RSA_4096",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ]
}
```

Pour plus d'informations sur l'utilisation de clés KMS asymétriques dans AWS KMS, voir Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés. Exemple 2 : pour convertir une clé publique au format DER (Linux et macOS)

L'`get-public-key`exemple suivant télécharge la clé publique d'une clé KMS asymétrique et l'enregistre dans un fichier DER.

Lorsque vous utilisez la `get-public-key` commande dans la AWS CLI, elle renvoie une clé publique X.509 codée DER codée en Base64. Cet exemple obtient la valeur de la `PublicKey` propriété sous forme de texte. Il les décode en Base64 `PublicKey` et les enregistre dans le fichier. `public_key.der` Le `output` paramètre renvoie la sortie sous forme de texte, au lieu de JSON. Le `--query` paramètre obtient uniquement la `PublicKey` propriété, et non les propriétés dont vous avez besoin pour utiliser la clé publique en toute sécurité en dehors de AWS KMS.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms get-public-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query PublicKey | base64 --decode > public_key.der
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur l'utilisation de clés KMS asymétriques dans AWS KMS, voir Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous GetPublicKeyà la section Référence des AWS CLI commandes.

`import-key-material`

L'exemple de code suivant montre comment utiliser`import-key-material`.

AWS CLI

Pour importer du matériel clé dans une clé KMS

L'`import-key-material`exemple suivant télécharge le contenu clé dans une clé KMS créée sans élément clé. L'état de la clé KMS doit être`PendingImport`.

Cette commande utilise des éléments clés que vous avez chiffrés avec la clé publique renvoyée par la `get-parameters-for-import` commande. Il utilise également le jeton d'importation issu de la même `get-parameters-for-import` commande.

Le `expiration-model` paramètre indique que le matériel clé expire automatiquement à la date et à l'heure spécifiées par le `valid-to` paramètre. Lorsque le contenu clé expire, AWS KMS le supprime, l'état de la clé KMS devient inutilisable. `Pending import` Pour restaurer la clé KMS, vous devez réimporter le même contenu clé. Pour utiliser un autre matériel clé, vous devez créer une nouvelle clé KMS.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé ou un ARN de clé valide provenant de votre AWS compte.

```
aws kms import-key-material \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2021-09-21T19:00:00Z
```

Sortie :

```
{
```

```
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

Pour plus d'informations sur l'importation de matériel clé, voir Importation de matériel clé dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ImportKeyMaterialà la section Référence des AWS CLI commandes.

`list-aliases`

L'exemple de code suivant montre comment utiliser`list-aliases`.

AWS CLI

Exemple 1 : pour répertorier tous les alias d'un AWS compte et d'une région

L'exemple suivant utilise la `list-aliases` commande pour répertorier tous les alias de la région par défaut du AWS compte. La sortie inclut les alias associés aux clés KMS AWS gérées et aux clés KMS gérées par le client.

```
aws kms list-aliases
```

Sortie :

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",
            "AliasName": "alias/testKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",
            "AliasName": "alias/FinanceDept",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "AliasName": "alias/aws/dynamodb",
```

```
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "AliasName": "alias/aws/ebs",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
        },
        ...
    ]
}
```

Exemple 2 : pour répertorier tous les alias d'une clé KMS particulière

L'exemple suivant utilise la `list-aliases` commande et son `key-id` paramètre pour répertorier tous les alias associés à une clé KMS particulière.

Chaque alias est associé à une seule clé KMS, mais une clé KMS peut avoir plusieurs alias. Cette commande est très utile car la console AWS KMS ne répertorie qu'un seul alias pour chaque clé KMS. Pour trouver tous les alias d'une clé KMS, vous devez utiliser la `list-aliases` commande.

Cet exemple utilise l'ID de clé KMS pour le `--key-id` paramètre, mais vous pouvez utiliser un ID de clé, un ARN de clé, un nom d'alias ou un ARN d'alias dans cette commande.

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "Aliases": [
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-key",
            "AliasName": "alias/oregon-test-key"
        },
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-test",
            "AliasName": "alias/project121-test"
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des alias dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListAliasesà la section Référence des AWS CLI commandes.

`list-grants`

L'exemple de code suivant montre comment utiliser`list-grants`.

AWS CLI

Pour afficher les subventions sur une clé AWS KMS

L'`list-grants`exemple suivant affiche toutes les autorisations associées à la clé KMS AWS gérée spécifiée pour Amazon DynamoDB dans votre compte. Cette autorisation permet à DynamoDB d'utiliser la clé KMS en votre nom pour chiffrer une table DynamoDB avant de l'écrire sur le disque. Vous pouvez utiliser une commande comme celle-ci pour afficher les autorisations relatives aux clés KMS AWS gérées et aux clés KMS gérées par le client dans le AWS compte et la région.

Cette commande utilise le `key-id` paramètre avec un ID de clé pour identifier la clé KMS. Vous pouvez utiliser un ID de clé ou un ARN de clé pour identifier la clé KMS. Pour obtenir l'ID de clé ou l'ARN d'une clé KMS AWS gérée, utilisez la `list-aliases` commande `list-keys` or.

```
aws kms list-grants \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Le résultat indique que l'autorisation autorise Amazon DynamoDB à utiliser la clé KMS pour des opérations cryptographiques, à consulter les informations relatives à la clé KMS `DescribeKey` () et à retirer les autorisations (). `RetireGrant` La `EncryptionContextSubset` contrainte limite ces autorisations aux demandes qui incluent les paires de contextes de chiffrement spécifiées. Par conséquent, les autorisations de la subvention ne sont effectives que sur le compte spécifié et sur la table DynamoDB.

```
{
    "Grants": [
        {
            "Constraints": {
                "EncryptionContextSubset": {
                    "aws:dynamodb:subscriberId": "123456789012",
```

```
                    "aws:dynamodb:tableName": "Services"
                }
            },
            "IssuingAccount": "arn:aws:iam::123456789012:root",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
            "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "CreationDate": "2021-05-13T18:32:45.144000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Subventions dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListGrantsà la section Référence des AWS CLI commandes.

`list-key-policies`

L'exemple de code suivant montre comment utiliser`list-key-policies`.

AWS CLI

Pour obtenir les noms des politiques clés pour une clé KMS

L'`list-key-policies`exemple suivant obtient les noms des politiques clés pour une clé gérée par le client dans les exemples de compte et de région. Vous pouvez utiliser cette commande pour rechercher les noms des politiques clés relatives aux clés AWS gérées et aux clés gérées par le client.

Comme le seul nom de politique clé valide est`default`, cette commande n'est pas utile.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise une valeur d'ID de clé, mais vous pouvez utiliser un ID de clé ou un ARN de clé dans cette commande.

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "PolicyNames": [
    "default"
    ]
}
```

Pour plus d'informations sur les politiques clés AWS KMS, consultez la section Utilisation des politiques clés dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListKeyPoliciesà la section Référence des AWS CLI commandes.

`list-key-rotations`

L'exemple de code suivant montre comment utiliser`list-key-rotations`.

AWS CLI

Pour récupérer des informations sur toutes les rotations de matériaux clés terminées

L'`list-key-rotations`exemple suivant répertorie les informations relatives à toutes les rotations de matériaux clés effectuées pour la clé KMS spécifiée.

```
aws kms list-key-rotations \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "Rotations": [
```

```
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "RotationDate": "2024-03-02T10:11:36.564000+00:00",
            "RotationType": "AUTOMATIC"
        },
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "RotationDate": "2024-04-05T15:14:47.757000+00:00",
            "RotationType": "ON_DEMAND"
        }
    ],
    "Truncated": false
}
```

Pour plus d'informations, voir Rotation des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListKeyRotationsà la section Référence des AWS CLI commandes.

`list-keys`

L'exemple de code suivant montre comment utiliser`list-keys`.

AWS CLI

Pour obtenir les clés KMS dans un compte et une région

L'`list-keys`exemple suivant permet d'obtenir les clés KMS d'un compte et d'une région. Cette commande renvoie à la fois les clés AWS gérées et les clés gérées par le client.

```
aws kms list-keys
```

Sortie :

```
{
    "Keys": [
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
```

```
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation des clés dans le guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListKeysà la section Référence des AWS CLI commandes.

`list-resource-tags`

L'exemple de code suivant montre comment utiliser`list-resource-tags`.

AWS CLI

Pour obtenir les balises d'une clé KMS

L'`list-resource-tags`exemple suivant permet d'obtenir les balises d'une clé KMS. Pour ajouter ou remplacer des balises de ressources sur les clés KMS, utilisez la `tag-resource` commande. La sortie indique que cette clé KMS possède deux balises de ressources, chacune ayant une clé et une valeur.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise une valeur d'ID de clé, mais vous pouvez utiliser un ID de clé ou un ARN de clé dans cette commande.

```
aws kms list-resource-tags \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "Tags": [
```

```
    {
        "TagKey": "Dept",
        "TagValue": "IT"
    },
    {
        "TagKey": "Purpose",
        "TagValue": "Test"
    }
    ],
    "Truncated": false
}
```

Pour plus d'informations sur l'utilisation des balises dans AWS KMS, consultez la section Balisage des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListResourceTagsà la section Référence des AWS CLI commandes.

`list-retirable-grants`

L'exemple de code suivant montre comment utiliser`list-retirable-grants`.

AWS CLI

Pour consulter les subventions qu'un directeur peut retirer

L'`list-retirable-grants`exemple suivant affiche toutes les autorisations que l'`ExampleAdmin`utilisateur peut retirer sur les clés KMS d'un AWS compte et d'une région. Vous pouvez utiliser une commande comme celle-ci pour afficher les autorisations que tout directeur de compte peut retirer sur les clés KMS du AWS compte et de la région.

La valeur du `retiring-principal` paramètre requis doit être le nom de ressource Amazon (ARN) d'un compte, d'un utilisateur ou d'un rôle.

Vous ne pouvez pas spécifier de service pour la valeur de `retiring-principal` dans cette commande, même si un service peut être le principal sortant. Pour trouver les subventions dans lesquelles un service particulier est le principal partant à la retraite, utilisez la `list-grants` commande.

Le résultat indique que `ExampleAdmin` l'utilisateur est autorisé à retirer les autorisations sur deux clés KMS différentes dans le compte et dans la région. Outre le capital retraité, le compte est autorisé à retirer toute subvention du compte.

```
aws kms list-retirable-grants \
    --retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

Sortie :

```
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "GrantId":
 "156b69c63cb154aa21f59929ff19760717be8d9d82b99df53e18b94a15a5e88e",
            "Name": "",
            "CreationDate": 2021-01-14T20:17:36.419000+00:00,
            "GranteePrincipal": "arn:aws:iam::111122223333:user/ExampleUser",
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Operations": [
                "Encrypt"
            ],
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            }
        },
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "GrantId":
 "8c94d1f12f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
            "Name": "",
            "CreationDate": "2021-02-02T19:49:49.638000+00:00",
            "GranteePrincipal": "arn:aws:iam::111122223333:role/ExampleRole",
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Operations": [
                "Decrypt"
            ],
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
```

```
            }
        }
    ],
    "Truncated": false
}
```

Pour plus d'informations, consultez la section Subventions dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ListRetirableGrantsà la section Référence des AWS CLI commandes.

`put-key-policy`

L'exemple de code suivant montre comment utiliser`put-key-policy`.

AWS CLI

Pour modifier la politique de clé d'une clé KMS

L'`put-key-policy`exemple suivant modifie la politique de clé pour une clé gérée par le client.

Pour commencer, créez une politique clé et enregistrez-la dans un fichier JSON local. Dans cet exemple, le fichier est`key_policy.json`. Vous pouvez également spécifier la politique clé sous forme de valeur de chaîne du `policy` paramètre.

La première déclaration de cette politique clé autorise le AWS compte à utiliser les politiques IAM pour contrôler l'accès à la clé KMS. La deuxième instruction autorise l'`test-user`utilisateur à exécuter les `list-keys` commandes `describe-key` et sur la clé KMS.

Contenu de `key_policy.json` :

```
{
    "Version" : "2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
```

```
            "Resource" : "*"
        },
        {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [
                "kms:DescribeKey",
                "kms:ListKeys"
            ],
            "Resource" : "*"
        }
    ]
}
```

Pour identifier la clé KMS, cet exemple utilise l'ID de la clé, mais vous pouvez également utiliser un ARN de clé. Pour spécifier la politique clé, la commande utilise le `policy` paramètre. Pour indiquer que la politique se trouve dans un fichier, elle utilise le `file://` préfixe requis. Ce préfixe est nécessaire pour identifier les fichiers sur tous les systèmes d'exploitation pris en charge. Enfin, la commande utilise le `policy-name` paramètre avec une valeur de`default`. Si aucun nom de politique n'est spécifié, la valeur par défaut est`default`. La seule valeur valide est `default`.

```
aws kms put-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --policy file://key_policy.json
```

Cette commande ne génère pas de sortie. Pour vérifier que la commande est efficace, `get-key-policy` utilisez-la. L'exemple de commande suivant permet d'obtenir la politique de clé pour la même clé KMS. Le `output` paramètre avec une valeur de `text` renvoie un format de texte facile à lire.

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text
```

Sortie :

```
{
    "Version" : "2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
            },
            {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [ "kms:Describe", "kms:List" ],
            "Resource" : "*"
        }
    ]
}
```

Pour plus d'informations, consultez la section Modification d'une politique clé dans le guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous PutKeyPolicyà la section Référence des AWS CLI commandes.

`re-encrypt`

L'exemple de code suivant montre comment utiliser`re-encrypt`.

AWS CLI

Exemple 1 : rechiffrer un message chiffré sous une autre clé KMS symétrique (Linux et macOS).

L'exemple de `re-encrypt` commande suivant illustre la méthode recommandée pour rechiffrer les données à l'aide de la CLI AWS .

Fournissez le texte chiffré dans un fichier. Dans la valeur du `--ciphertext-blob` paramètre, utilisez le `fileb://` préfixe, qui indique à la CLI de lire les données d'un fichier binaire. Si le fichier ne se trouve pas dans le répertoire actuel, saisissez le chemin complet du fichier. Pour plus d'informations sur la lecture des valeurs des paramètres de la AWS CLI à partir d'un fichier, consultez la section Chargement de paramètres AWS CLI depuis un fichier dans le guide de l'utilisateur de l'interface de ligne de AWS commande et les meilleures pratiques relatives aux paramètres de fichiers locaux dans le blog des outils de ligne de AWS commande. Spécifiez la clé KMS source, qui déchiffre le texte chiffré. Le `--source-key-id` paramètre n'est pas obligatoire lors du déchiffrement à l'aide de clés KMS de chiffrement symétriques. AWS KMS peut obtenir la clé KMS qui a été utilisée pour chiffrer les données à partir des métadonnées du blob de texte chiffré. Toutefois, la spécification de la clé KMS que vous utilisez est une bonne pratique. Cette pratique garantit que vous utilisez la clé KMS comme vous le souhaitez et vous empêche de déchiffrer par inadvertance un texte chiffré à l'aide d'une clé KMS non fiable. Spécifiez la clé KMS de destination, qui chiffre à nouveau les données. Le paramètre est toujours obligatoire. `--destination-key-id` Cet exemple utilise un ARN de clé, mais vous pouvez utiliser n'importe quel identifiant de clé valide. Demandez la sortie en texte brut sous forme de valeur de texte.Le `--query` paramètre indique à la CLI de n'obtenir que la valeur du champ à partir de `Plaintext` la sortie. Le `--output` paramètre renvoie la sortie sous forme de texte.Base64 décodez le texte en clair et enregistrez-le dans un fichier. L'exemple suivant montre comment rediriger (|) la valeur du `Plaintext` paramètre vers l'utilitaire Base64, qui le décode. Ensuite, il redirige (>) la sortie décodée vers le `ExamplePlaintext` fichier.

Avant d'exécuter cette commande, remplacez l'exemple de clé par IDs des identifiants de clé valides provenant de votre AWS compte.

```
aws kms re-encrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --query CiphertextBlob \
    --output text | base64 --decode > ExampleReEncryptedFile
```

Cette commande ne produit aucun résultat. La sortie de la `re-encrypt` commande est décodée en base64 et enregistrée dans un fichier.

Pour plus d'informations, consultez la référence ReEncryptde l'API du service de gestion des AWS clés.

Exemple 2 : pour rechiffrer un message chiffré sous une autre clé KMS symétrique (invite de commande Windows).

L'exemple de `re-encrypt` commande suivant est identique au précédent, sauf qu'il utilise l'`certutil`utilitaire pour décoder les données en texte brut en Base64. Cette procédure nécessite deux commandes, comme indiqué dans les exemples suivants.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms re-encrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^
    --query CiphertextBlob ^
    --output text > ExampleReEncryptedFile.base64
```

Utilisez ensuite l'`certutil`utilitaire

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Sortie :

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Pour plus d'informations, consultez la référence ReEncryptde l'API du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ReEncryptà la section Référence des AWS CLI commandes.

`retire-grant`

L'exemple de code suivant montre comment utiliser`retire-grant`.

AWS CLI

Pour annuler une subvention sur la clé principale d'un client

L'`retire-grant`exemple suivant supprime une autorisation d'une clé KMS.

L'exemple de commande suivant spécifie les `key-id` paramètres `grant-id` et. La valeur du `key-id` paramètre doit être l'ARN de la clé KMS.

```
aws kms retire-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne produit aucun résultat. Pour confirmer que la subvention a été retirée, utilisez la `list-grants` commande.

Pour plus d'informations, consultez la section Retrait et révocation des subventions dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous RetireGrantà la section Référence des AWS CLI commandes.

`revoke-grant`

L'exemple de code suivant montre comment utiliser`revoke-grant`.

AWS CLI

Pour révoquer une autorisation sur la clé principale d'un client

L'`revoke-grant`exemple suivant supprime une autorisation d'une clé KMS. L'exemple de commande suivant spécifie les `key-id` paramètres `grant-id` et. La valeur du `key-id` paramètre peut être l'ID de clé ou l'ARN de la clé KMS.

```
aws kms revoke-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne produit aucun résultat. Pour confirmer que l'autorisation a été révoquée, utilisez la `list-grants` commande.

Pour plus d'informations, consultez la section Retrait et révocation des subventions dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous RevokeGrantà la section Référence des AWS CLI commandes.

`rotate-key-on-demand`

L'exemple de code suivant montre comment utiliser`rotate-key-on-demand`.

AWS CLI

Pour effectuer la rotation à la demande d'une clé KMS

L'`rotate-key-on-demand`exemple suivant lance immédiatement la rotation du matériau clé pour la clé KMS spécifiée.

```
aws kms rotate-key-on-demand \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Sortie :

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Pour plus d'informations, consultez la section Comment effectuer une rotation des clés à la demande dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous RotateKeyOnDemandà la section Référence des AWS CLI commandes.

`schedule-key-deletion`

L'exemple de code suivant montre comment utiliser`schedule-key-deletion`.

AWS CLI

Pour planifier la suppression d'une clé KMS gérée par le client.

L'`schedule-key-deletion`exemple suivant planifie la suppression de la clé KMS gérée par le client spécifiée dans les 15 jours.

Le `--key-id` paramètre identifie la clé KMS. Cet exemple utilise une valeur ARN clé, mais vous pouvez utiliser l'ID de clé ou l'ARN de la clé KMS. Le `--pending-window-in-days` paramètre indique la durée de la période d'attente de 7 à 30 jours. Par défaut, le délai d'attente est de 30 jours. Cet exemple indique une valeur de 15, qui indique de AWS supprimer définitivement la clé KMS 15 jours après la fin de la commande.

```
aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --pending-window-in-days 15
```

La réponse inclut l'ARN de la clé, l'état de la clé, le délai d'attente (`PendingWindowInDays`) et la date de suppression en heure Unix. Pour afficher la date de suppression en heure locale, utilisez la console AWS KMS. Les clés KMS à l'état `PendingDeletion` clé ne peuvent pas être utilisées dans des opérations cryptographiques.

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": "2022-06-18T23:43:51.272000+00:00",
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 15
}
```

Pour plus d'informations, consultez la section Suppression de clés dans le guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous ScheduleKeyDeletionà la section Référence des AWS CLI commandes.

`sign`

L'exemple de code suivant montre comment utiliser`sign`.

AWS CLI

Exemple 1 : pour générer une signature numérique pour un message

L'`sign`exemple suivant génère une signature cryptographique pour un message court. La sortie de la commande inclut un `Signature` champ codé en base 64 que vous pouvez vérifier à l'aide de la `verify` commande.

Vous devez spécifier un message à signer et un algorithme de signature pris en charge par votre clé KMS asymétrique. Pour obtenir les algorithmes de signature de votre clé KMS, utilisez la `describe-key` commande.

Dans la AWS CLI v2, la valeur du `message` paramètre doit être codée en Base64. Vous pouvez également enregistrer le message dans un fichier et utiliser le `fileb://` préfixe, qui indique à la AWS CLI de lire les données binaires du fichier.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte. L'ID de clé doit représenter une clé KMS asymétrique avec une clé d'utilisation de SIGN_VERIFY.

```
msg=(echo 'Hello World' | base64)

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://UnsignedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Pour plus d'informations sur l'utilisation de clés KMS asymétriques dans AWS KMS, voir Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Exemple 2 : pour enregistrer une signature numérique dans un fichier (Linux et macOS)

L'`sign`exemple suivant génère une signature cryptographique pour un message court stocké dans un fichier local. La commande obtient également la `Signature` propriété de la réponse, la décode en Base64 et l'enregistre dans le fichier. ExampleSignature Vous pouvez utiliser le fichier de signature dans une `verify` commande qui vérifie la signature.

La `sign` commande nécessite un message codé en Base64 et un algorithme de signature pris en charge par votre clé KMS asymétrique. Pour obtenir les algorithmes de signature pris en charge par votre clé KMS, utilisez la `describe-key` commande.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte. L'ID de clé doit représenter une clé KMS asymétrique avec une clé d'utilisation de SIGN_VERIFY.

```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature
```

Cette commande ne produit aucun résultat. Cet exemple extrait la `Signature` propriété de la sortie et l'enregistre dans un fichier.

Pour plus d'informations sur l'utilisation de clés KMS asymétriques dans AWS KMS, voir Clés asymétriques dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, consultez la section Référence des AWS CLI commandes de connexion.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une étiquette à une clé KMS

L'`tag-resource`exemple suivant ajoute `"Purpose":"Test"` des `"Dept":"IT"` balises à une clé KMS gérée par le client. Vous pouvez utiliser de telles balises pour étiqueter les clés KMS et créer des catégories de clés KMS à des fins d'autorisation et d'audit.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise une valeur d'ID de clé, mais vous pouvez utiliser un ID de clé ou un ARN de clé dans cette commande.

```
aws kms tag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

Cette commande ne produit aucun résultat. Pour afficher les balises d'une clé AWS KMS KMS, utilisez la `list-resource-tags` commande.

Pour plus d'informations sur l'utilisation des balises dans AWS KMS, consultez la section Balisage des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une étiquette d'une clé KMS

L'`untag-resource`exemple suivant supprime la balise contenant la `"Purpose"` clé d'une clé KMS gérée par le client.

Pour spécifier la clé KMS, utilisez le `key-id` paramètre. Cet exemple utilise une valeur d'ID de clé, mais vous pouvez utiliser un ID de clé ou un ARN de clé dans cette commande. Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms untag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --tag-key 'Purpose'
```

Cette commande ne produit aucun résultat. Pour afficher les balises d'une clé AWS KMS KMS, utilisez la `list-resource-tags` commande.

Pour plus d'informations sur l'utilisation des balises dans AWS KMS, consultez la section Balisage des clés dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-alias`

L'exemple de code suivant montre comment utiliser`update-alias`.

AWS CLI

Pour associer un alias à une autre clé KMS

L'`update-alias`exemple suivant associe l'alias `alias/test-key` à une autre clé KMS.

Le `--alias-name` paramètre spécifie l'alias. La valeur du nom d'alias doit commencer par `alias/` .Le `--target-key-id` paramètre spécifie la clé KMS à associer à l'alias. Il n'est pas nécessaire de spécifier la clé KMS actuelle pour l'alias.

```
aws kms update-alias \
    --alias-name alias/test-key \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette commande ne produit aucun résultat. Pour trouver l'alias, utilisez la `list-aliases` commande.

Pour plus d'informations, consultez la section Mise à jour des alias dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous UpdateAliasà la section Référence des AWS CLI commandes.

`update-custom-key-store`

L'exemple de code suivant montre comment utiliser`update-custom-key-store`.

AWS CLI

Exemple 1 : pour modifier le nom convivial d'un magasin de clés personnalisé

L'`update-custom-key-store`exemple suivant modifie le nom du magasin de clés personnalisé. Cet exemple fonctionne pour un magasin de clés AWS CloudHSM ou un magasin de clés externe.

Utilisez le `custom-key-store-id` pour identifier le magasin de clés. Utilisez le `new-custom-key-store-name` paramètre pour spécifier le nouveau nom convivial.

Pour mettre à jour le nom convivial d'un magasin de clés AWS CloudHSM, vous devez d'abord le déconnecter, par exemple à l'aide de la commande. `disconnect-custom-key-store` Vous pouvez mettre à jour le nom convivial d'un magasin de clés externe lorsqu'il est connecté ou

déconnecté. Pour connaître l'état de connexion de votre magasin de clés personnalisé, utilisez la `describe-custom-key-store` commande.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --new-custom-key-store-name ExampleKeyStore
```

Cette commande ne renvoie aucune donnée. Pour vérifier que la commande a fonctionné, utilisez une `describe-custom-key-stores` commande.

Pour plus d'informations sur la mise à jour d'un magasin de clés AWS CloudHSM, consultez la section Modification des paramètres du magasin de clés AWS CloudHSM dans le guide du développeur du service de gestion des clés.AWS

Pour plus d'informations sur la mise à jour d'un magasin de clés externe, consultez la section Modification des propriétés du magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

Exemple 2 : pour modifier le mot de passe kmsuser d'un magasin de clés CloudHSM AWS

L'`update-custom-key-store`exemple suivant met à jour la valeur du `kmsuser` mot de passe par le mot de passe actuel du cluster CloudHSM associé au magasin de clés spécifié. `kmsuser` Cette commande ne modifie pas le `kmsuser` mot de passe du cluster. Il indique simplement à AWS KMS le mot de passe actuel. Si KMS ne possède pas le mot de `kmsuser` passe actuel, il ne peut pas se connecter au magasin de AWS clés CloudHSM.

REMARQUE : Avant de mettre à jour un magasin de clés AWS CloudHSM, vous devez le déconnecter. Utilisez la commande `disconnect-custom-key-store`. Une fois la commande terminée, vous pouvez reconnecter le magasin de clés AWS CloudHSM. Utilisez la commande `connect-custom-key-store`.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --key-store-password ExamplePassword
```

Cette commande ne renvoie aucune sortie. Pour vérifier que la modification a été effective, utilisez une `describe-custom-key-stores` commande.

Pour plus d'informations sur la mise à jour d'un magasin de clés AWS CloudHSM, consultez la section Modification des paramètres du magasin de clés AWS CloudHSM dans le guide du développeur du service de gestion des clés.AWS

Exemple 3 : pour modifier le cluster AWS CloudHSM d'un AWS magasin de clés CloudHSM

L'exemple suivant remplace le cluster AWS CloudHSM associé à un magasin de clés AWS CloudHSM par un cluster apparenté, tel qu'une sauvegarde différente du même cluster.

REMARQUE : Avant de mettre à jour un magasin de clés AWS CloudHSM, vous devez le déconnecter. Utilisez la commande `disconnect-custom-key-store`. Une fois la commande terminée, vous pouvez reconnecter le magasin de clés AWS CloudHSM. Utilisez la commande `connect-custom-key-store`.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

Cette commande ne renvoie aucune sortie. Pour vérifier que la modification a été effective, utilisez une `describe-custom-key-stores` commande.

Pour plus d'informations sur la mise à jour d'un magasin de clés AWS CloudHSM, consultez la section Modification des paramètres du magasin de clés AWS CloudHSM dans le guide du développeur du service de gestion des clés.AWS

Exemple 4 : pour modifier les informations d'authentification du proxy d'un magasin de clés externe

L'exemple suivant met à jour les informations d'identification d'authentification du proxy pour votre banque de clés externe. Vous devez spécifier à la fois le `raw-secret-access-key` et le`access-key-id`, même si vous ne modifiez qu'une seule des valeurs. Vous pouvez utiliser cette fonctionnalité pour corriger un identifiant non valide ou pour le modifier lorsque le proxy externe du magasin de clés le fait pivoter.

Établissez les informations d'identification d'authentification proxy pour AWS KMS sur votre magasin de clés externe. Utilisez ensuite cette commande pour fournir les informations d'identification à AWS KMS. AWS KMS utilise ces informations d'identification pour signer ses demandes à votre proxy de stockage de clés externe.

Vous pouvez mettre à jour les informations d'authentification du proxy lorsque le magasin de clés externe est connecté ou déconnecté. Pour connaître l'état de connexion de votre magasin de clés personnalisé, utilisez la `describe-custom-key-store` commande.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Cette commande ne renvoie aucune sortie. Pour vérifier que la modification a été effective, utilisez une `describe-custom-key-stores` commande.

Pour plus d'informations sur la mise à jour d'un magasin de clés externe, consultez la section Modification des propriétés du magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

Exemple 5 : pour modifier la connectivité proxy d'un magasin de clés externe

Dans l'exemple suivant, l'option de connectivité proxy du magasin de clés externe passe de la connectivité du point de terminaison public à la connectivité du service de point de terminaison VPC. Outre la modification de la `xks-proxy-connectivity` valeur, vous devez modifier la `xks-proxy-uri-endpoint` valeur pour refléter le nom DNS privé associé au service de point de terminaison VPC. Vous devez également ajouter une `xks-proxy-vpc-endpoint-service-name` valeur.

REMARQUE : Avant de mettre à jour la connectivité proxy d'un magasin externe, vous devez le déconnecter. Utilisez la commande `disconnect-custom-key-store`. Une fois la commande terminée, vous pouvez reconnecter le magasin de clés externe à l'aide de la `connect-custom-key-store` commande.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
    --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \
    --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-
example"
```

Cette commande ne renvoie aucune sortie. Pour vérifier que la modification a été effective, utilisez une `describe-custom-key-stores` commande.

Pour plus d'informations sur la mise à jour d'un magasin de clés externe, consultez la section Modification des propriétés du magasin de clés externe dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous UpdateCustomKeyStoreà la section Référence des AWS CLI commandes.

`update-key-description`

L'exemple de code suivant montre comment utiliser`update-key-description`.

AWS CLI

Exemple 1 : pour ajouter ou modifier une description à une clé KMS gérée par le client

L'`update-key-description`exemple suivant ajoute une description à une clé KMS gérée par le client. Vous pouvez utiliser la même commande pour modifier une description existante.

Le `--key-id` paramètre identifie la clé KMS dans la commande. Cet exemple utilise une valeur d'ARN clé, mais vous pouvez utiliser l'ID de clé ou l'ARN clé de la clé KMS. Le `--description` paramètre spécifie la nouvelle description. La valeur de ce paramètre remplace la description actuelle de la clé KMS, le cas échéant.

```
aws kms update-key-description \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --description "IT Department test key"
```

Cette commande ne produit aucun résultat. Pour afficher la description d'une clé KMS, utilisez la `describe-key` commande.

Pour plus d'informations, consultez la référence UpdateKeyDescriptionde l'API du service de gestion des AWS clés.

Exemple 2 : pour supprimer la description d'une clé KMS gérée par le client

L'`update-key-description`exemple suivant supprime la description d'une clé KMS gérée par le client.

Le `--key-id` paramètre identifie la clé KMS dans la commande. Cet exemple utilise une valeur d'ID de clé, mais vous pouvez utiliser l'ID de clé ou l'ARN de la clé KMS. Le `--description` paramètre avec une valeur de chaîne vide («) supprime la description existante.

```
aws kms update-key-description \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --description ''
```

Cette commande ne produit aucun résultat. Pour afficher la description d'une clé KMS, utilisez la commande describe-key.

Pour plus d'informations, consultez la référence UpdateKeyDescriptionde l'API du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous UpdateKeyDescriptionà la section Référence des AWS CLI commandes.

`verify-mac`

L'exemple de code suivant montre comment utiliser`verify-mac`.

AWS CLI

Exemple 1 : pour vérifier un HMAC

La `verify-mac` commande suivante vérifie un HMAC pour un message spécifique, des clés HMAC KMS et un algorithme MAC. La valeur « true » dans la MacValid réponse indique que le HMAC est valide.

Dans la AWS CLI v2, la valeur du `message` paramètre doit être codée en Base64. Vous pouvez également enregistrer le message dans un fichier et utiliser le `fileb://` préfixe, qui indique à la AWS CLI de lire les données binaires du fichier.

Le MAC que vous spécifiez ne peut pas être codé en base64. Pour obtenir de l'aide pour décoder le MAC renvoyé par la `generate-mac` commande, consultez les exemples de `generate-mac` commandes.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte. L'ID de clé doit représenter une clé HMAC KMS dont l'utilisation de `GENERATE_VERIFY_MAC` la clé est.

```
msg=(echo 'Hello World' | base64)

aws kms verify-mac \
```

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://Message \
    --mac-algorithm HMAC_SHA_384 \
    --mac fileb://ExampleMac
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "MacValid": true,
    "MacAlgorithm": "HMAC_SHA_384"
}
```

Pour plus d'informations sur l'utilisation des clés HMAC KMS dans AWS KMS, consultez la section Clés HMAC dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, reportez-vous VerifyMacà la section Référence des AWS CLI commandes.

`verify`

L'exemple de code suivant montre comment utiliser`verify`.

AWS CLI

Pour vérifier une signature numérique

La `verify` commande suivante vérifie la signature cryptographique d'un court message codé en Base64. L'identifiant de clé, le message, le type de message et l'algorithme de signature doivent être identiques à ceux utilisés pour signer le message.

Dans la AWS CLI v2, la valeur du `message` paramètre doit être codée en Base64. Vous pouvez également enregistrer le message dans un fichier et utiliser le `fileb://` préfixe, qui indique à la AWS CLI de lire les données binaires du fichier.

La signature que vous spécifiez ne peut pas être codée en base64. Pour obtenir de l'aide pour décoder la signature renvoyée par la `sign` commande, consultez les exemples de `sign` commandes.

La sortie de la commande inclut un `SignatureValid` champ booléen qui indique que la signature a été vérifiée. Si la validation de signature échoue, la `verify` commande échoue également.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un ID de clé valide provenant de votre AWS compte.

```
aws kms verify \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --signature fileb://ExampleSignature
```

Sortie :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "SignatureValid": true,
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Pour plus d'informations sur l'utilisation de clés KMS asymétriques dans AWS KMS, consultez la section Utilisation de clés asymétriques dans le Guide du développeur du service de gestion des AWS clés.

- Pour plus de détails sur l'API, voir Vérifier dans AWS CLI la référence des commandes.

Exemples de Lake Formation utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de la Formation AWS Command Line Interface with Lake.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-lf-tags-to-resource`

L'exemple de code suivant montre comment utiliser`add-lf-tags-to-resource`.

AWS CLI

Pour associer une ou plusieurs balises LF à une ressource existante

L'`add-lf-tags-to-resource`exemple suivant attache une balise LF donnée à la ressource de table.

```
aws lakeformation add-lf-tags-to-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "analyst"
        ]
    }]
}
```

Sortie :

```
{
```

```
    "Failures": []
}
```

Pour plus d'informations, consultez la section Affectation de balises LF aux ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous AddLfTagsToResourceà la section Référence des AWS CLI commandes.

`batch-grant-permissions`

L'exemple de code suivant montre comment utiliser`batch-grant-permissions`.

AWS CLI

Pour accorder des autorisations groupées sur les ressources aux principaux

L'`batch-grant-permissions`exemple suivant accorde l'accès groupé à des ressources spécifiées aux principaux.

```
aws lakeformation batch-grant-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Entries": [{
            "Id": "1",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
            ],
```

```
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
        {
            "Id": "2",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_customer"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
        {
            "Id": "3",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
```

```
        {
            "Id": "4",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "DataCellsFilter": {
                    "TableCatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "TableName": "dl_tpc_item",
                    "Name": "developer_item"
                }
            },
            "Permissions": [
                "SELECT"
            ],
            "PermissionsWithGrantOption": []
        }
    ]
}
```

Sortie :

```
{
    "Failures": []
}
```

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous BatchGrantPermissionsà la section Référence des AWS CLI commandes.

`batch-revoke-permissions`

L'exemple de code suivant montre comment utiliser`batch-revoke-permissions`.

AWS CLI

Pour révoquer en bloc les autorisations sur les ressources accordées aux principaux

L'`batch-revoke-permissions`exemple suivant révoque en bloc l'accès aux ressources spécifiées aux principaux.

```
aws lakeformation batch-revoke-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Entries": [{
            "Id": "1",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
        {
            "Id": "2",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
```

```
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        }
    ]
}
```

Sortie :

```
{
    "Failures": []
}
```

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous BatchRevokePermissionsà la section Référence des AWS CLI commandes.

`cancel-transaction`

L'exemple de code suivant montre comment utiliser`cancel-transaction`.

AWS CLI

Pour annuler une transaction

L'`cancel-transaction`exemple suivant annule la transaction.

```
aws lakeformation cancel-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CancelTransactionà la section Référence des AWS CLI commandes.

`commit-transaction`

L'exemple de code suivant montre comment utiliser`commit-transaction`.

AWS CLI

Pour valider la transaction

L'`commit-transaction`exemple suivant valide la transaction.

```
aws lakeformation commit-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Sortie :

```
{
    "TransactionStatus": "committed"
}
```

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CommitTransactionà la section Référence des AWS CLI commandes.

`create-data-cells-filter`

L'exemple de code suivant montre comment utiliser`create-data-cells-filter`.

AWS CLI

Exemple 1 : pour créer un filtre de cellule de données

L'`create-data-cells-filter`exemple suivant crée un filtre de cellule de données pour autoriser l'accès à certaines colonnes en fonction de la condition des lignes.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "TableData": {
        "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],
        "DatabaseName": "tpc",
        "Name": "developer_promotion",
        "RowFilter": {
            "FilterExpression": "p_promo_name='ese'"
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Filtrage des données et sécurité au niveau des cellules dans Lake Formation in the AWS Lake Formation Developer Guide.

Exemple 2 : pour créer un filtre de colonne

L'`create-data-cells-filter`exemple suivant crée un filtre de données pour autoriser l'accès à certaines colonnes.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "TableData": {
        "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],
        "DatabaseName": "tpc",
        "Name": "developer_promotion_allrows",
        "RowFilter": {
            "AllRowsWildcard": {}
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Filtrage des données et sécurité au niveau des cellules dans Lake Formation in the AWS Lake Formation Developer Guide.

Exemple 3 : pour créer un filtre de données avec des colonnes d'exclusion

L'`create-data-cells-filter`exemple suivant crée un filtre de données pour autoriser l'accès à toutes les colonnes sauf aux colonnes mentionnées.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "TableData": {
        "ColumnWildcard": {
            "ExcludedColumnNames": ["p_channel_details", "p_start_date_sk"]
        },
        "DatabaseName": "tpc",
        "Name": "developer_promotion_excludecolumn",
        "RowFilter": {
            "AllRowsWildcard": {}
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Filtrage des données et sécurité au niveau des cellules dans Lake Formation in the AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateDataCellsFilterà la section Référence des AWS CLI commandes.

`create-lf-tag`

L'exemple de code suivant montre comment utiliser`create-lf-tag`.

AWS CLI

Pour créer un LF-Tag

L'`create-lf-tag`exemple suivant crée une balise LF avec le nom et les valeurs spécifiés.

```
aws lakeformation create-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup' \
    --tag-values '["developer","analyst","campaign"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises LF pour le contrôle d'accès aux métadonnées dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous CreateLfTagà la section Référence des AWS CLI commandes.

`delete-data-cells-filter`

L'exemple de code suivant montre comment utiliser`delete-data-cells-filter`.

AWS CLI

Pour supprimer le filtre de cellules de données

L'`delete-data-cells-filter`exemple suivant supprime un filtre de cellule de données donné.

```
aws lakeformation delete-data-cells-filter \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "TableCatalogId": "123456789111",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_promotion",
    "Name": "developer_promotion"
```

```
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Filtrage des données et sécurité au niveau des cellules dans Lake Formation in the AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteDataCellsFilterà la section Référence des AWS CLI commandes.

`delete-lf-tag`

L'exemple de code suivant montre comment utiliser`delete-lf-tag`.

AWS CLI

Pour supprimer la définition de la balise LF

L'`delete-lf-tag`exemple suivant supprime la définition de la balise LF.

```
aws lakeformation delete-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises LF pour le contrôle d'accès aux métadonnées dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous DeleteLfTagà la section Référence des AWS CLI commandes.

`delete-objects-on-cancel`

L'exemple de code suivant montre comment utiliser`delete-objects-on-cancel`.

AWS CLI

Pour supprimer un objet lorsque la transaction est annulée

L'`delete-objects-on-cancel`exemple suivant supprime l'objet s3 répertorié lorsque la transaction est annulée.

```
aws lakeformation delete-objects-on-cancel \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
    "TransactionId": "1234d972ca8347b89825e33c5774aec4",
    "Objects": [{
        "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
        "ETag": "1234ab1fc50a316b149b4e1f21a73800"
    }]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsOnCancelà la section Référence des AWS CLI commandes.

`deregister-resource`

L'exemple de code suivant montre comment utiliser`deregister-resource`.

AWS CLI

Pour désenregistrer le stockage des lacs de données

L'`deregister-resource`exemple suivant annule l'enregistrement de la ressource telle qu'elle est gérée par la Lake Formation.

```
aws lakeformation deregister-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout d'un emplacement Amazon S3 à votre lac de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous DeregisterResourceà la section Référence des AWS CLI commandes.

`describe-transaction`

L'exemple de code suivant montre comment utiliser`describe-transaction`.

AWS CLI

Pour récupérer les détails d'une transaction

L'`describe-transaction`exemple suivant renvoie les détails d'une seule transaction.

```
aws lakeformation describe-transaction \
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Sortie :

```
{
    "TransactionDescription": {
        "TransactionId": "12345972ca8347b89825e33c5774aec4",
        "TransactionStatus": "committed",
        "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
        "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
    }
}
```

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeTransactionà la section Référence des AWS CLI commandes.

`extend-transaction`

L'exemple de code suivant montre comment utiliser`extend-transaction`.

AWS CLI

Pour prolonger une transaction

L'`extend-transaction`exemple suivant étend la transaction.

```
aws lakeformation extend-transaction \
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ExtendTransactionà la section Référence des AWS CLI commandes.

`get-data-lake-settings`

L'exemple de code suivant montre comment utiliser`get-data-lake-settings`.

AWS CLI

Pour récupérer les paramètres AWS du lac de données géré par Lake Formation

L'`get-data-lake-settings`exemple suivant extrait la liste des administrateurs du lac de données et d'autres paramètres du lac de données.

```
aws lakeformation get-data-lake-settings \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111"
}
```

Sortie :

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [{
            "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"
        }],
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": [
                {
                        "Principal": {
                            "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                        },
                        "Permissions": [
                            "ALL"
                        ]
                }
        ],
        "TrustedResourceOwners": [],
        "AllowExternalDataFiltering": true,
        "ExternalDataFilteringAllowList": [{
            "DataLakePrincipalIdentifier": "123456789111"
        }],
        "AuthorizedSessionTagValueList": [
            "Amazon EMR"
        ]
    }
}
```

Pour plus d'informations, consultez la section Modification des paramètres de sécurité par défaut pour votre lac de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetDataLakeSettingsà la section Référence des AWS CLI commandes.

`get-effective-permissions-for-path`

L'exemple de code suivant montre comment utiliser`get-effective-permissions-for-path`.

AWS CLI

Pour récupérer des autorisations sur des ressources situées sur un chemin spécifique

L'`get-effective-permissions-for-path`exemple suivant renvoie les autorisations de Lake Formation pour une table ou une ressource de base de données spécifiée située sur un chemin dans Amazon S3.

```
aws lakeformation get-effective-permissions-for-path \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
}
```

Sortie :

```
{
    "Permissions": [{
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": []
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/EMR-
RuntimeRole"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
```

```
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": []
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:saml-
provider/oktaSAMLProvider:user/emr-developer"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "ALL",
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": []
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "ALL",
                "ALTER",
                "CREATE_TABLE",
                "DESCRIBE",
                "DROP"
            ],
            "PermissionsWithGrantOption": [
                "ALL",
                "ALTER",
                "CREATE_TABLE",
                "DESCRIBE",
                "DROP"
```

```
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/LF-
GlueServiceRole"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "CREATE_TABLE"
            ],
            "PermissionsWithGrantOption": []
        }
    ],
    "NextToken":
 "E5SlJDSTZleUp6SWpvaU9UQTNORE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWlP
}
```

Pour plus d'informations, consultez la section Gestion des autorisations relatives à la formation des lacs dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetEffectivePermissionsForPathà la section Référence des AWS CLI commandes.

`get-lf-tag`

L'exemple de code suivant montre comment utiliser`get-lf-tag`.

AWS CLI

Pour récupérer la définition du tag LF

L'`get-lf-tag`exemple suivant récupère la définition de la balise LF.

```
aws lakeformation get-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup'
```

Sortie :

```
{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
        "analyst",
        "campaign",
        "developer"
    ]
}
```

Pour plus d'informations, consultez la section Gestion des balises LF pour le contrôle d'accès aux métadonnées dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetLfTagà la section Référence des AWS CLI commandes.

`get-query-state`

L'exemple de code suivant montre comment utiliser`get-query-state`.

AWS CLI

Pour récupérer l'état d'une requête soumise

L'`get-query-state`exemple suivant renvoie l'état d'une requête soumise précédemment.

```
aws lakeformation get-query-state \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Sortie :

```
{
    "State": "FINISHED"
}
```

Pour plus d'informations, consultez la section Opérations de données transactionnelles dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetQueryStateà la section Référence des AWS CLI commandes.

`get-query-statistics`

L'exemple de code suivant montre comment utiliser`get-query-statistics`.

AWS CLI

Pour récupérer les statistiques des requêtes

L'`get-query-statistics`exemple suivant permet de récupérer des statistiques sur la planification et l'exécution d'une requête.

```
aws lakeformation get-query-statistics \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Sortie :

```
{
    "ExecutionStatistics": {
        "AverageExecutionTimeMillis": 0,
        "DataScannedBytes": 0,
        "WorkUnitsExecutedCount": 0
    },
    "PlanningStatistics": {
        "EstimatedDataToScanBytes": 43235,
        "PlanningTimeMillis": 2377,
        "QueueTimeMillis": 440,
        "WorkUnitsGeneratedCount": 1
    },
    "QuerySubmissionTime": "2022-08-11T02:14:38.641870+00:00"
}
```

Pour plus d'informations, consultez la section Opérations de données transactionnelles dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetQueryStatisticsà la section Référence des AWS CLI commandes.

`get-resource-lf-tags`

L'exemple de code suivant montre comment utiliser`get-resource-lf-tags`.

AWS CLI

Pour répertorier les balises LF

L'`list-lf-tags`exemple suivant renvoie la liste des balises LF que le demandeur est autorisé à consulter.

```
aws lakeformation list-lf-tags \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "ResourceShareType": "ALL",
    "MaxResults": 2
}
```

Sortie :

```
{
"LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "category",
        "TagValues": [
            "private",
            "public"
        ]
    },
    {
        "CatalogId": "123456789111",
        "TagKey": "group",
        "TagValues": [
            "analyst",
            "campaign",
            "developer"
        ]
    }],
    "NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWNvbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"
}
```

Pour plus d'informations, consultez la section Gestion des balises LF pour le contrôle d'accès aux métadonnées dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetResourceLfTagsà la section Référence des AWS CLI commandes.

`get-table-objects`

L'exemple de code suivant montre comment utiliser`get-table-objects`.

AWS CLI

Pour répertorier les objets d'une table gouvernée

L'`get-table-objects`exemple suivant renvoie l'ensemble des objets Amazon S3 qui constituent la table gouvernée spécifiée.

```
aws lakeformation get-table-objects \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
    "QueryAsOfTime": "2022-08-10T15:00:00"
}
```

Sortie :

```
{
    "Objects": [{
        "PartitionValues": [],
        "Objects": [{
            "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
            "ETag": "12345b1fc50a316b149b4e1f21a73800",
            "Size": 43235
        }]
```

```
    }]
}
```

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetTableObjectsà la section Référence des AWS CLI commandes.

`get-work-unit-results`

L'exemple de code suivant montre comment utiliser`get-work-unit-results`.

AWS CLI

Pour récupérer les unités de travail d'une requête donnée

L'`get-work-unit-results`exemple suivant renvoie les unités de travail résultant de la requête.

```
aws lakeformation get-work-units \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \
    --work-unit-id '0' \
    --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

Sortie :

```
outfile with Blob content.
```

Pour plus d'informations, consultez la section Opérations de données transactionnelles dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetWorkUnitResultsà la section Référence des AWS CLI commandes.

`get-work-units`

L'exemple de code suivant montre comment utiliser`get-work-units`.

AWS CLI

Pour récupérer des unités de travail

L'`get-work-units`exemple suivant récupère les unités de travail générées par l' StartQueryPlanning opération.

```
aws lakeformation get-work-units \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Sortie :

```
{
    "WorkUnitRanges": [{
        "WorkUnitIdMax": 0,
        "WorkUnitIdMin": 0,
        "WorkUnitToken":
 "1234eMAk4kLO4umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
+f88jzGrYq22gE6jkQlpOB
+Oet2eqNUmFudAAAAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMCOEWRda
wAAAAEAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmBl
+VSojiGOtdsUk7vcjYXUUboYm3dvqRqX2s4gROMOn
+Ij8R0/8jYmnHkpvyAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytmOhozxDOEi/
F2ZoXz6wlGDfGUo/2WxCkYOhTyNaw6TM
+7drTM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzBlAjBH8okRtYP8R7AY2Wls/
hqFBhG0V4l42AC0LxsuZbMQrE2SzWZUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCNklExMR
+btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
    }],
    "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
}
```

Pour plus d'informations, consultez la section Opérations de données transactionnelles dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GetWorkUnitsà la section Référence des AWS CLI commandes.

`grant-permissions`

L'exemple de code suivant montre comment utiliser`grant-permissions`.

AWS CLI

Exemple 1 : accorder des autorisations au principal sur les ressources à l'aide de balises LF

L'`grant-permissions`exemple suivant accorde TOUTES les autorisations au principal sur la ressource de base de données qui correspond à la politique LF-Tag.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"
    },
    "Resource": {
        "LFTagPolicy": {
            "CatalogId": "123456789111",
            "ResourceType": "DATABASE",
            "Expression": [{
                "TagKey": "usergroup",
                "TagValues": [
                    "analyst",
                    "developer"
                ]
            }]
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": [
        "ALL"
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

Exemple 2 : Pour accorder des autorisations au niveau des colonnes au principal

L'`grant-permissions`exemple suivant autorise le principal à sélectionner une colonne spécifique.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "TableWithColumns": {
            "CatalogId": "123456789111",
            "ColumnNames": ["p_end_date_sk"],
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "Permissions": [
        "SELECT"
    ],
    "PermissionsWithGrantOption": []
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

Exemple 3 : Pour accorder des autorisations de table au principal

L'`grant-permissions`exemple suivant accorde au principal l'autorisation de sélection sur toutes les tables d'une base de données donnée.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Principal": {
```

```
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableWildcard": {}
        }
    },
    "Permissions": [
        "SELECT"
    ],
    "PermissionsWithGrantOption": []
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

Exemple 4 : Pour accorder des autorisations sur les balises LF au principal

L'`grant-permissions`exemple suivant accorde une autorisation d'association sur les balises LF au principal.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private", "public"
            ]
        }
```

```
    },
    "Permissions": [
        "ASSOCIATE"
    ],
    "PermissionsWithGrantOption": []
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

Exemple 5 : Pour accorder des autorisations sur l'emplacement des données au principal

L'`grant-permissions`exemple suivant autorise le principal à localiser les données.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "DataLocation": {
            "CatalogId": "123456789111",
            "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
        }
    },
    "Permissions": [
        "DATA_LOCATION_ACCESS"
    ],
    "PermissionsWithGrantOption": []
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous GrantPermissionsà la section Référence des AWS CLI commandes.

`list-data-cells-filter`

L'exemple de code suivant montre comment utiliser`list-data-cells-filter`.

AWS CLI

Pour répertorier les filtres de cellules de données

L'`list-data-cells-filter`exemple suivant répertorie les filtres de cellules de données pour une table donnée.

```
aws lakeformation list-data-cells-filter \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "MaxResults": 2,
    "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
    }
}
```

Sortie :

```
{
    "DataCellsFilters": [{
            "TableCatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableName": "dl_tpc_promotion",
            "Name": "developer_promotion",
            "RowFilter": {
                "FilterExpression": "p_promo_name='ese'"
            },
            "ColumnNames": [
                "p_channel_details",
```

```
                "p_start_date_sk",
                "p_purpose",
                "p_promo_id",
                "p_promo_name",
                "p_end_date_sk",
                "p_discount_active"
            ]
        },
        {
            "TableCatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableName": "dl_tpc_promotion",
            "Name": "developer_promotion_allrows",
            "RowFilter": {
                "FilterExpression": "TRUE",
                "AllRowsWildcard": {}
            },
            "ColumnNames": [
                "p_channel_details",
                "p_start_date_sk",
                "p_promo_name"
            ]
        }
    ],
    "NextToken": "2MDA2MTgwNiwibmFub3MiOjE0MDAwMDAwMH19"
}
```

Pour plus d'informations, voir Filtrage des données et sécurité au niveau des cellules dans Lake Formation in the AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDataCellsFilterà la section Référence des AWS CLI commandes.

`list-permissions`

L'exemple de code suivant montre comment utiliser`list-permissions`.

AWS CLI

Exemple 1 : pour récupérer la liste des principales autorisations sur la ressource

L'`list-permissions`exemple suivant renvoie une liste des principales autorisations sur les ressources de base de données.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "ResourceType": "DATABASE",
    "MaxResults": 2
}
```

Sortie :

```
{
    "PrincipalResourcePermissions": [{
        "Principal": {
            "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
        },
        "Resource": {
            "Database": {
                "CatalogId": "123456789111",
                "Name": "tpc"
            }
        },
        "Permissions": [
            "DESCRIBE"
        ],
        "PermissionsWithGrantOption": []
    }],
    "NextToken":
 "E5SlJDSTZleUp6SWpvaU9UQTNORE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWlP
}
```

Pour plus d'informations, consultez la section Gestion des autorisations relatives à la formation des lacs dans le guide du développeur de AWS Lake Formation.

Exemple 2 : Pour récupérer la liste des principales autorisations sur la table à l'aide de filtres de données

L'`list-permissions`exemple suivant répertorie les autorisations sur le tableau avec les filtres de données associés accordés au principal.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_customer"
        }
    },
    "IncludeRelated": "TRUE",
    "MaxResults": 10
}
```

Sortie :

```
{
    "PrincipalResourcePermissions": [{
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "customer",
                    "Name": "customer_invoice"
                }
            },
            "Permissions": [
                "ALL",
                "ALTER",
                "DELETE",
                "DESCRIBE",
                "DROP",
                "INSERT"
            ],
            "PermissionsWithGrantOption": [
```

```
                "ALL",
                "ALTER",
                "DELETE",
                "DESCRIBE",
                "DROP",
                "INSERT"
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
            },
            "Resource": {
                "TableWithColumns": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "customer",
                    "Name": "customer_invoice",
                    "ColumnWildcard": {}
                }
            },
            "Permissions": [
                "SELECT"
            ],
            "PermissionsWithGrantOption": [
                "SELECT"
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
            },
            "Resource": {
                "DataCellsFilter": {
                    "TableCatalogId": "123456789111",
                    "DatabaseName": "customer",
                    "TableName": "customer_invoice",
                    "Name": "dl_us_customer"
                }
            },
            "Permissions": [
                "DESCRIBE",
                "SELECT",
```

```
                "DROP"
            ],
            "PermissionsWithGrantOption": []
        }
    ],
    "NextToken": "VyeUFjY291bnRQZXJtaXNzaW9ucyI6ZmFsc2V9"
}
```

Pour plus d'informations, consultez la section Gestion des autorisations relatives à la formation des lacs dans le guide du développeur de AWS Lake Formation.

Exemple 3 : Pour récupérer la liste des principales autorisations sur les balises LF

L'`list-permissions`exemple suivant répertorie les autorisations sur les balises LF accordées au principal.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private"
            ]
        }
    },
    "MaxResults": 10
}
```

Sortie :

```
{
    "PrincipalResourcePermissions": [{
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
```

```
            },
            "Resource": {
                "LFTag": {
                    "CatalogId": "123456789111",
                    "TagKey": "category",
                    "TagValues": [
                        "*"
                    ]
                }
            },
            "Permissions": [
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": [
                "DESCRIBE"
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            },
            "Resource": {
                "LFTag": {
                    "CatalogId": "123456789111",
                    "TagKey": "category",
                    "TagValues": [
                        "*"
                    ]
                }
            },
            "Permissions": [
                "ASSOCIATE"
            ],
            "PermissionsWithGrantOption": [
                "ASSOCIATE"
            ]
        }
    ],
    "NextToken": "EJwY21GMGFXOXVJanA3SW5Ocm1pc3Npb25zIjpmYWxzZX0="
}
```

Pour plus d'informations, consultez la section Gestion des autorisations relatives à la formation des lacs dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous ListPermissionsà la section Référence des AWS CLI commandes.

`list-resources`

L'exemple de code suivant montre comment utiliser`list-resources`.

AWS CLI

Pour répertorier les ressources gérées par la Lake Formation

L'`list-resources`exemple suivant répertorie les ressources correspondant à la condition qui est gérée par la Lake Formation.

```
aws lakeformation list-resources \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "FilterConditionList": [{
        "Field": "ROLE_ARN",
        "ComparisonOperator": "CONTAINS",
        "StringValueList": [
            "123456789111"
        ]
    }],
    "MaxResults": 10
}
```

Sortie :

```
{
    "ResourceInfoList": [{
            "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111",
            "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole",
            "LastModified": "2022-07-21T02:12:46.669000+00:00"
        },
        {
```

```
            "ResourceArn": "arn:aws:s3:::lf-emr-test-123456789111",
            "RoleArn": "arn:aws:iam::123456789111:role/EMRLFS3Role",
            "LastModified": "2022-07-29T16:22:03.211000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des autorisations relatives à la formation des lacs dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous ListResourcesà la section Référence des AWS CLI commandes.

`list-transactions`

L'exemple de code suivant montre comment utiliser`list-transactions`.

AWS CLI

Pour répertorier tous les détails des transactions

L'`list-transactions`exemple suivant renvoie des métadonnées relatives aux transactions et à leur statut.

```
aws lakeformation list-transactions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "StatusFilter": "ALL",
    "MaxResults": 3
}
```

Sortie :

```
{
    "Transactions": [{
            "TransactionId": "1234569f08804cb790d950d4d0fe485e",
            "TransactionStatus": "committed",
            "TransactionStartTime": "2022-08-10T14:32:29.220000+00:00",
```

```
            "TransactionEndTime": "2022-08-10T14:32:33.751000+00:00"
        },
        {
            "TransactionId": "12345972ca8347b89825e33c5774aec4",
            "TransactionStatus": "committed",
            "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
            "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
        },
        {
            "TransactionId": "12345daf6cb047dbba8ad9b0414613b2",
            "TransactionStatus": "committed",
            "TransactionStartTime": "2022-08-10T13:56:51.261000+00:00",
            "TransactionEndTime": "2022-08-10T13:56:51.547000+00:00"
        }
    ],
    "NextToken": "77X1ebypsI7os+X2lhHsZLGNCDK3nNGpwRdFpicSOHgcX1/
QMoniUAKcpR3kj3ts3PVdMA=="
}
```

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTransactionsà la section Référence des AWS CLI commandes.

`put-data-lake-settings`

L'exemple de code suivant montre comment utiliser`put-data-lake-settings`.

AWS CLI

Pour définir les paramètres AWS du lac de données géré par Lake Formation

L'`put-data-lake-settings`exemple suivant définit la liste des administrateurs du lac de données et les autres paramètres du lac de données.

```
aws lakeformation put-data-lake-settings \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
```

```
    "DataLakeSettings": {
        "DataLakeAdmins": [{
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            }
        ],
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": [],
        "TrustedResourceOwners": [],
        "AllowExternalDataFiltering": true,
        "ExternalDataFilteringAllowList": [{
            "DataLakePrincipalIdentifier ": "123456789111"
        }],
        "AuthorizedSessionTagValueList": ["Amazon EMR"]
    }
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Modification des paramètres de sécurité par défaut pour votre lac de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous PutDataLakeSettingsà la section Référence des AWS CLI commandes.

`register-resource`

L'exemple de code suivant montre comment utiliser`register-resource`.

AWS CLI

Exemple 1 : pour enregistrer le stockage d'un lac de données à l'aide d'un rôle lié au service

L'`register-resource`exemple suivant enregistre la ressource comme étant gérée par la Lake Formation à l'aide du rôle lié au service.

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
```

```
    "UseServiceLinkedRole": true
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout d'un emplacement Amazon S3 à votre lac de données dans le guide du développeur de AWS Lake Formation.

Exemple 2 : pour enregistrer le stockage des lacs de données à l'aide d'un rôle personnalisé

L'`register-resource`exemple suivant enregistre la ressource comme étant gérée par la Lake Formation à l'aide d'un rôle personnalisé.

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
    "UseServiceLinkedRole": false,
    "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout d'un emplacement Amazon S3 à votre lac de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous RegisterResourceà la section Référence des AWS CLI commandes.

`remove-lf-tags-from-resource`

L'exemple de code suivant montre comment utiliser`remove-lf-tags-from-resource`.

AWS CLI

Pour supprimer le tag LF d'une ressource

L'`remove-lf-tags-from-resource`exemple suivant supprime l'association LF-tag avec la ressource de table.

```
aws lakeformation remove-lf-tags-from-resource \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

Sortie :

```
{
    "Failures": []
}
```

Pour plus d'informations, consultez la section Affectation de balises LF aux ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous RemoveLfTagsFromResourceà la section Référence des AWS CLI commandes.

`revoke-permissions`

L'exemple de code suivant montre comment utiliser`revoke-permissions`.

AWS CLI

Pour révoquer les autorisations sur les ressources accordées au principal

L'`revoke-permissions`exemple suivant révoque l'accès principal à une table spécifique d'une base de données donnée.

```
aws lakeformation revoke-permissions \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": []
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous RevokePermissionsà la section Référence des AWS CLI commandes.

`search-databases-by-lf-tags`

L'exemple de code suivant montre comment utiliser`search-databases-by-lf-tags`.

AWS CLI

Pour effectuer une recherche dans les ressources de la base de données par LFTags

L'`search-databases-by-lf-tags`exemple suivant permet de rechercher des ressources de base de données correspondant à LFTag une expression.

```
aws lakeformation search-databases-by-lf-tags \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "MaxResults": 1,
    "CatalogId": "123456789111",
    "Expression": [{
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

Sortie :

```
{
    "DatabaseList": [{
        "Database": {
            "CatalogId": "123456789111",
            "Name": "tpc"
        },
        "LFTags": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
                "developer"
            ]
        }]
    }]
}
```

Pour plus d'informations, consultez la section Affichage des ressources auxquelles un tag LF est attribué dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous SearchDatabasesByLfTagsà la section Référence des AWS CLI commandes.

`search-tables-by-lf-tags`

L'exemple de code suivant montre comment utiliser`search-tables-by-lf-tags`.

AWS CLI

Pour effectuer une recherche dans les ressources du tableau par LFTags

L'`search-tables-by-lf-tags`exemple suivant montre comment rechercher une LFTag expression correspondant aux ressources d'une table.

```
aws lakeformation search-tables-by-lf-tags \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "MaxResults": 2,
    "CatalogId": "123456789111",
    "Expression": [{
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

Sortie :

```
{
    "NextToken": "c2VhcmNoQWxsVGFnc0luVGFibGVzIjpmYWxzZX0=",
    "TableList": [{
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_item"
        },
        "LFTagOnDatabase": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
                "developer"
            ]
```

```
        }],
        "LFTagsOnTable": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
                "developer"
            ]
        }],
        "LFTagsOnColumns": [{
                "Name": "i_item_desc",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_container",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_wholesale_cost",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_manufact_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
```

```
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_brand_id",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_formulation",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_current_price",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_size",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
```

```
            {
                "Name": "i_rec_start_date",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_manufact",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_item_sk",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_manager_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_item_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
```

```
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_class_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_class",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_category",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_category_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
```

```
            }]
        },
        {
            "Name": "i_brand",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_units",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_rec_end_date",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_color",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_product_name",
```

```
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            }
        ]
    }]
}
```

Pour plus d'informations, consultez la section Affichage des ressources auxquelles un tag LF est attribué dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous SearchTablesByLfTagsà la section Référence des AWS CLI commandes.

`start-query-planning`

L'exemple de code suivant montre comment utiliser`start-query-planning`.

AWS CLI

Pour traiter l'instruction de requête

L'`start-query-planning`exemple suivant envoie une demande pour traiter une instruction de requête.

```
aws lakeformation start-query-planning \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "QueryPlanningContext": {
        "CatalogId": "012345678901",
        "DatabaseName": "tpc"
    },
    "QueryString": "select * from dl_tpc_household_demographics_gov where
 hd_income_band_sk=9"
}
```

Sortie :

```
{
    "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"
}
```

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartQueryPlanningà la section Référence des AWS CLI commandes.

`start-transaction`

L'exemple de code suivant montre comment utiliser`start-transaction`.

AWS CLI

Pour démarrer une nouvelle transaction

L'`start-transaction`exemple suivant lance une nouvelle transaction et renvoie son numéro de transaction.

```
aws lakeformation start-transaction \
    --transaction-type = 'READ_AND_WRITE'
```

Sortie :

```
{
    "TransactionId": "b014d972ca8347b89825e33c5774aec4"
}
```

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartTransactionà la section Référence des AWS CLI commandes.

`update-lf-tag`

L'exemple de code suivant montre comment utiliser`update-lf-tag`.

AWS CLI

Pour mettre à jour la définition du tag LF

L'`update-lf-tag`exemple suivant met à jour la définition de la balise LF.

```
aws lakeformation update-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup' \
    --tag-values-to-add '["admin"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des balises LF pour le contrôle d'accès aux métadonnées dans le guide du développeur de AWS Lake Formation.

- Pour plus de détails sur l'API, reportez-vous UpdateLfTagà la section Référence des AWS CLI commandes.

`update-table-objects`

L'exemple de code suivant montre comment utiliser`update-table-objects`.

AWS CLI

Pour modifier les objets d'une table gouvernée

L'`update-table-objects`exemple suivant ajoute les objets S3 fournis à la table gouvernée spécifiée.

```
aws lakeformation update-table-objects \
    --cli-input-json file://input.json
```

Contenu de `input.json` :

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
    "TransactionId": "12347a9f75424b9b915f6ff201d2a190",
    "WriteOperations": [{
        "AddObject": {
```

```
            "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
            "ETag": "1234ab1fc50a316b149b4e1f21a73800",
            "Size": 42200
        }
    }]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Lire et écrire dans le lac de données dans le cadre de transactions dans le AWS Lake Formation Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateTableObjectsà la section Référence des AWS CLI commandes.

Exemples Lambda utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Lambda.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-layer-version-permission`

L'exemple de code suivant montre comment utiliser`add-layer-version-permission`.

AWS CLI

Pour ajouter des autorisations à une version de couche

L'`add-layer-version-permission`exemple suivant autorise le compte spécifié à utiliser la version 1 de la couche`my-layer`.

```
aws lambda add-layer-version-permission \
    --layer-name my-layer \
    --statement-id xaccount \
    --action lambda:GetLayerVersion  \
    --principal 123456789012 \
    --version-number 1
```

Sortie :

```
{
    "RevisionId": "35d87451-f796-4a3f-a618-95a3671b0a0c",
    "Statement":
    {
        "Sid":"xaccount",
        "Effect":"Allow",
        "Principal":{
            "AWS":"arn:aws:iam::210987654321:root"
        },
        "Action":"lambda:GetLayerVersion",
        "Resource":"arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1"
    }
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous AddLayerVersionPermissionà la section Référence des AWS CLI commandes.

`add-permission`

L'exemple de code suivant montre comment utiliser`add-permission`.

AWS CLI

Pour ajouter des autorisations à une fonction Lambda existante

L'`add-permission`exemple suivant accorde au service Amazon SNS l'autorisation d'appeler une fonction nommée. `my-function`

```
aws lambda add-permission \
    --function-name my-function \
    --action lambda:InvokeFunction \
    --statement-id sns \
    --principal sns.amazonaws.com
```

Sortie :

```
{
    "Statement":
    {
        "Sid":"sns",
        "Effect":"Allow",
        "Principal":{
            "Service":"sns.amazonaws.com"
        },
        "Action":"lambda:InvokeFunction",
        "Resource":"arn:aws:lambda:us-east-2:123456789012:function:my-function"
    }
}
```

Pour plus d'informations, consultez la section Utilisation de politiques basées sur les ressources pour Lambda AWS dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous AddPermissionà la section Référence des AWS CLI commandes.

`create-alias`

L'exemple de code suivant montre comment utiliser`create-alias`.

AWS CLI

Pour créer un alias de fonction Lambda

L'exemple `create-alias` suivant crée un alias de fonction Lambda nommé `LIVE` qui pointe vers la version 1 de la fonction Lambda `my-function`.

```
aws lambda create-alias \
    --function-name my-function \
    --description "alias for live version of function" \
```

```
    --function-version 1 \
    --name LIVE
```

Sortie :

```
{
    "FunctionVersion": "1",
    "Name": "LIVE",
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
    "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",
    "Description": "alias for live version of function"
}
```

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous CreateAliasà la section Référence des AWS CLI commandes.

`create-event-source-mapping`

L'exemple de code suivant montre comment utiliser`create-event-source-mapping`.

AWS CLI

Pour créer un mappage entre une source d'événements et une fonction AWS Lambda

L'`create-event-source-mapping`exemple suivant crée un mappage entre une file d'attente SQS et la fonction `my-function` Lambda.

```
aws lambda create-event-source-mapping \
    --function-name my-function \
    --batch-size 5 \
    --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

Sortie :

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569284520.333,
```

```
    "BatchSize": 5,
    "State": "Creating",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

Pour plus d'informations, consultez la section Mappage des sources d'événements AWS Lambda dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous CreateEventSourceMappingà la section Référence des AWS CLI commandes.

`create-function`

L'exemple de code suivant montre comment utiliser`create-function`.

AWS CLI

Pour créer une fonction Lambda

L'exemple `create-function` suivant crée une fonction Lambda nommée `my-function`.

```
aws lambda create-function \
    --function-name my-function \
    --runtime nodejs18.x \
    --zip-file fileb://my-function.zip \
    --handler my-function.handler \
    --role arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-tges6bf4
```

Contenu de `my-function.zip` :

```
This file is a deployment package that contains your function code and any
 dependencies.
```

Sortie :

```
{
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "PFn4S+er27qk+UuZSTKEQfNKG/XNn7QJs90mJgq6oH8=",
    "FunctionName": "my-function",
```

```
    "CodeSize": 308,
    "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",
    "MemorySize": 128,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Version": "$LATEST",
    "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-
zgur6bf4",
    "Timeout": 3,
    "LastModified": "2023-10-14T22:26:11.234+0000",
    "Handler": "my-function.handler",
    "Runtime": "nodejs18.x",
    "Description": ""
}
```

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous CreateFunctionà la section Référence des AWS CLI commandes.

`delete-alias`

L'exemple de code suivant montre comment utiliser`delete-alias`.

AWS CLI

Pour supprimer un alias de fonction Lambda

L'exemple `delete-alias` suivant supprime l'alias nommé `LIVE` de la fonction Lambda `my-function`.

```
aws lambda delete-alias \
    --function-name my-function \
    --name LIVE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous DeleteAliasà la section Référence des AWS CLI commandes.

`delete-event-source-mapping`

L'exemple de code suivant montre comment utiliser`delete-event-source-mapping`.

AWS CLI

Pour supprimer le mappage entre une source d'événement et une fonction AWS Lambda

L'`delete-event-source-mapping`exemple suivant supprime le mappage entre une file d'attente SQS et la fonction Lambda`my-function`.

```
aws lambda delete-event-source-mapping \
    --uuid  a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569285870.271,
    "BatchSize": 5,
    "State": "Deleting",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

Pour plus d'informations, consultez la section Mappage des sources d'événements AWS Lambda dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous DeleteEventSourceMappingà la section Référence des AWS CLI commandes.

`delete-function-concurrency`

L'exemple de code suivant montre comment utiliser`delete-function-concurrency`.

AWS CLI

Pour supprimer une limite d’exécution simultanée d’une fonction

L’exemple `delete-function-concurrency` suivant supprime la limite d’exécution simultanée réservée de la fonction `my-function`.

```
aws lambda delete-function-concurrency \
    --function-name  my-function
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Réservation de simultanéité pour une fonction Lambda dans le Guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous DeleteFunctionConcurrencyà la section Référence des AWS CLI commandes.

`delete-function-event-invoke-config`

L'exemple de code suivant montre comment utiliser`delete-function-event-invoke-config`.

AWS CLI

Pour supprimer une configuration d'appel asynchrone

L'`delete-function-event-invoke-config`exemple suivant supprime la configuration d'appel asynchrone pour l'`GREEN`alias de la fonction spécifiée.

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- Pour plus de détails sur l'API, reportez-vous DeleteFunctionEventInvokeConfigà la section Référence des AWS CLI commandes.

`delete-function`

L'exemple de code suivant montre comment utiliser`delete-function`.

AWS CLI

Exemple 1 : pour supprimer une fonction Lambda par nom de fonction

L'exemple `delete-function` suivant supprime la fonction Lambda `my-function` nommée en spécifiant le nom de la fonction.

```
aws lambda delete-function \
    --function-name my-function
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour supprimer une fonction Lambda par ARN de fonction

L'exemple `delete-function` suivant supprime la fonction Lambda nommée `my-function` en spécifiant l'ARN de la fonction.

```
aws lambda delete-function \
    --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Cette commande ne produit aucun résultat.

Exemple 3 : pour supprimer une fonction Lambda par ARN partiel de fonction

L'exemple `delete-function` suivant supprime la fonction Lambda nommée `my-function` en spécifiant l'ARN partiel de la fonction.

```
aws lambda delete-function \
    --function-name 123456789012:function:my-function
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous DeleteFunctionà la section Référence des AWS CLI commandes.

`delete-layer-version`

L'exemple de code suivant montre comment utiliser`delete-layer-version`.

AWS CLI

Pour supprimer une version d'une couche Lambda

L'`delete-layer-version`exemple suivant supprime la version 2 de la couche nommée`my-layer`.

```
aws lambda delete-layer-version \
    --layer-name my-layer \
```

```
    --version-number 2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous DeleteLayerVersionà la section Référence des AWS CLI commandes.

`delete-provisioned-concurrency-config`

L'exemple de code suivant montre comment utiliser`delete-provisioned-concurrency-config`.

AWS CLI

Pour supprimer une configuration de simultanéité allouée

L'exemple `delete-provisioned-concurrency-config` suivant supprime la configuration de simultanéité allouée pour l'alias `GREEN` de la fonction spécifiée.

```
aws lambda delete-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier GREEN
```

- Pour plus de détails sur l'API, reportez-vous DeleteProvisionedConcurrencyConfigà la section Référence des AWS CLI commandes.

`get-account-settings`

L'exemple de code suivant montre comment utiliser`get-account-settings`.

AWS CLI

Pour récupérer les informations relatives à votre compte dans une AWS région

L'exemple `get-account-settings` suivant affiche les limites Lambda et les informations d'utilisation de votre compte.

```
aws lambda get-account-settings
```

Sortie :

```
{
    "AccountLimit": {
       "CodeSizeUnzipped": 262144000,
       "UnreservedConcurrentExecutions": 1000,
       "ConcurrentExecutions": 1000,
       "CodeSizeZipped": 52428800,
       "TotalCodeSize": 80530636800
    },
    "AccountUsage": {
       "FunctionCount": 4,
       "TotalCodeSize": 9426
    }
}
```

Pour plus d'informations, consultez Limites AWS Lambda dans le Guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous GetAccountSettingsà la section Référence des AWS CLI commandes.

`get-alias`

L'exemple de code suivant montre comment utiliser`get-alias`.

AWS CLI

Pour récupérer les informations relatives à un alias de fonction

L'exemple `get-alias` suivant affiche les détails de l'alias nommé `LIVE` de la fonction Lambda `my-function`.

```
aws lambda get-alias \
    --function-name my-function \
    --name LIVE
```

Sortie :

```
{
    "FunctionVersion": "3",
```

```
    "Name": "LIVE",
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
    "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",
    "Description": "alias for live version of function"
}
```

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous GetAliasà la section Référence des AWS CLI commandes.

`get-event-source-mapping`

L'exemple de code suivant montre comment utiliser`get-event-source-mapping`.

AWS CLI

Pour récupérer des informations sur le mappage d'une source d'événement

L'`get-event-source-mapping`exemple suivant affiche les détails du mappage entre une file d'attente SQS et la fonction `my-function` Lambda.

```
aws lambda get-event-source-mapping \
    --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Sortie :

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569284520.333,
    "BatchSize": 5,
    "State": "Enabled",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

Pour plus d'informations, consultez la section Mappage des sources d'événements AWS Lambda dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous GetEventSourceMappingà la section Référence des AWS CLI commandes.

`get-function-concurrency`

L'exemple de code suivant montre comment utiliser`get-function-concurrency`.

AWS CLI

Pour afficher le paramètre de simultanéité réservé pour une fonction

L'exemple `get-function-concurrency` suivant récupère le paramètre de simultanéité réservé pour la fonction spécifiée.

```
aws lambda get-function-concurrency \
    --function-name my-function
```

Sortie :

```
{
    "ReservedConcurrentExecutions": 250
}
```

- Pour plus de détails sur l'API, reportez-vous GetFunctionConcurrencyà la section Référence des AWS CLI commandes.

`get-function-configuration`

L'exemple de code suivant montre comment utiliser`get-function-configuration`.

AWS CLI

Pour récupérer les paramètres d'une fonction Lambda spécifiques à sa version

L'exemple `get-function-configuration` suivant affiche les paramètres de la version 2 de la fonction `my-function`.

```
aws lambda get-function-configuration \
    --function-name  my-function:2
```

Sortie :

```
{
    "FunctionName": "my-function",
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
    "MemorySize": 256,
    "Version": "2",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",
    "Handler": "index.handler"
}
```

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous GetFunctionConfigurationà la section Référence des AWS CLI commandes.

`get-function-event-invoke-config`

L'exemple de code suivant montre comment utiliser`get-function-event-invoke-config`.

AWS CLI

Pour afficher une configuration d'appel asynchrone

L'`get-function-event-invoke-config`exemple suivant récupère la configuration d'appel asynchrone pour l'BLUEalias de la fonction spécifiée.

```
aws lambda get-function-event-invoke-config \
```

```
    --function-name my-function:BLUE
```

Sortie :

```
{
    "LastModified": 1577824396.653,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {
            "Destination": "arn:aws:sqs:us-east-2:123456789012:failed-invocations"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFunctionEventInvokeConfigà la section Référence des AWS CLI commandes.

`get-function`

L'exemple de code suivant montre comment utiliser`get-function`.

AWS CLI

Pour récupérer des informations sur une fonction

L'exemple `get-function` suivant affiche des informations sur la fonction `my-function`.

```
aws lambda get-function \
    --function-name  my-function
```

Sortie :

```
{
    "Concurrency": {
        "ReservedConcurrentExecutions": 100
    },
    "Code": {
```

```
        "RepositoryType": "S3",
        "Location": "https://awslambda-us-west-2-tasks.s3.us-west-2.amazonaws.com/
snapshots/123456789012/my-function..."
    },
    "Configuration": {
        "TracingConfig": {
            "Mode": "PassThrough"
        },
        "Version": "$LATEST",
        "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",
        "FunctionName": "my-function",
        "VpcConfig": {
            "SubnetIds": [],
            "VpcId": "",
            "SecurityGroupIds": []
        },
        "MemorySize": 128,
        "RevisionId": "28f0fb31-5c5c-43d3-8955-03e76c5c1075",
        "CodeSize": 304,
        "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
        "Handler": "index.handler",
        "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-role-
uy3l9qyq",
        "Timeout": 3,
        "LastModified": "2019-09-24T18:20:35.054+0000",
        "Runtime": "nodejs10.x",
        "Description": ""
    }
}
```

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous GetFunctionà la section Référence des AWS CLI commandes.

`get-layer-version-by-arn`

L'exemple de code suivant montre comment utiliser`get-layer-version-by-arn`.

AWS CLI

Pour récupérer des informations sur une version de couche Lambda

L'`get-layer-version-by-arn`exemple suivant affiche des informations sur la version de la couche avec le nom de ressource Amazon (ARN) spécifié.

```
aws lambda get-layer-version-by-arn \
    --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2"
```

Sortie :

```
{
    "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2",
    "Description": "AWS Lambda SciPy layer for Python 3.11 (scipy-1.1.0, numpy-1.15.4) https://github.com/scipy/scipy/releases/tag/v1.1.0 https://github.com/numpy/numpy/releases/tag/v1.15.4",
    "CreatedDate": "2023-10-12T10:09:38.398+0000",
    "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x",
    "Content": {
        "CodeSize": 41784542,
        "CodeSha256": "GGmv8ocUw4cly0T8HL0Vx/f5V4RmSCGNjDIslY4VskM=",
        "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/snapshots/123456789012/..."
    },
    "Version": 2,
    "CompatibleRuntimes": [
        "python3.11"
    ],
    "LicenseInfo": "SciPy: https://github.com/scipy/scipy/blob/main/LICENSE.txt, NumPy: https://github.com/numpy/numpy/blob/main/LICENSE.txt"
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous GetLayerVersionByArnà la section Référence des AWS CLI commandes.

`get-layer-version-policy`

L'exemple de code suivant montre comment utiliser`get-layer-version-policy`.

AWS CLI

Pour récupérer la politique d'autorisations pour une version de couche Lambda

L'`get-layer-version-policy`exemple suivant affiche les informations de politique relatives à la version 1 pour la couche nomméemy`-layer`.

```
aws lambda get-layer-version-policy \
    --layer-name my-layer \
    --version-number 1
```

Sortie :

```
{
    "Policy": {
        "Version":"2012-10-17",
        "Id":"default",
        "Statement":
        [
            {
                "Sid":"xaccount",
                "Effect":"Allow",
                "Principal": {"AWS":"arn:aws:iam::123456789012:root"},
                "Action":"lambda:GetLayerVersion",
                "Resource":"arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1"
            }
        ]
    },
    "RevisionId": "c68f21d2-cbf0-4026-90f6-1375ee465cd0"
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous GetLayerVersionPolicyà la section Référence des AWS CLI commandes.

`get-layer-version`

L'exemple de code suivant montre comment utiliser`get-layer-version`.

AWS CLI

Pour récupérer des informations sur une version de couche Lambda

L'`get-layer-version`exemple suivant affiche les informations relatives à la version 1 de la couche nomméemy-`layer`.

```
aws lambda get-layer-version \
    --layer-name my-layer \
    --version-number 1
```

Sortie :

```
{
    "Content": {
        "Location": "https://awslambda-us-east-2-layers.s3.us-east-2.amazonaws.com/
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?
versionId=27iWyA73cCAYqyH...",
        "CodeSha256": "tv9jJO+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",
        "CodeSize": 169
    },
    "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
    "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1",
    "Description": "My Python layer",
    "CreatedDate": "2018-11-14T23:03:52.894+0000",
    "Version": 1,
    "LicenseInfo": "MIT",
    "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
    ]
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous GetLayerVersionà la section Référence des AWS CLI commandes.

`get-policy`

L'exemple de code suivant montre comment utiliser`get-policy`.

AWS CLI

Pour récupérer la politique IAM basée sur les ressources d'une fonction, d'une version ou d'un alias

L'exemple `get-policy` suivant affiche des informations de politique relative à la fonction Lambda `my-function`.

```
aws lambda get-policy \
    --function-name my-function
```

Sortie :

```
{
    "Policy": {
        "Version":"2012-10-17",
        "Id":"default",
        "Statement":
        [
            {
                "Sid":"iot-events",
                "Effect":"Allow",
                "Principal": {"Service":"iotevents.amazonaws.com"},
                "Action":"lambda:InvokeFunction",
                "Resource":"arn:aws:lambda:us-west-2:123456789012:function:my-
function"
            }
        ]
    },
    "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668"
}
```

Pour plus d'informations, consultez la section Utilisation de politiques basées sur les ressources pour Lambda AWS dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous GetPolicyà la section Référence des AWS CLI commandes.

`get-provisioned-concurrency-config`

L'exemple de code suivant montre comment utiliser`get-provisioned-concurrency-config`.

AWS CLI

Pour afficher la configuration de la simultanéité allouée

L'exemple `get-provisioned-concurrency-config` suivant affiche les détails de la configuration de simultanéité allouée pour l'alias BLUE de la fonction spécifiée.

```
aws lambda get-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier BLUE
```

Sortie :

```
{
    "RequestedProvisionedConcurrentExecutions": 100,
    "AvailableProvisionedConcurrentExecutions": 100,
    "AllocatedProvisionedConcurrentExecutions": 100,
    "Status": "READY",
    "LastModified": "2019-12-31T20:28:49+0000"
}
```

- Pour plus de détails sur l'API, reportez-vous GetProvisionedConcurrencyConfigà la section Référence des AWS CLI commandes.

`invoke`

L'exemple de code suivant montre comment utiliser`invoke`.

AWS CLI

Exemple 1 : pour invoquer une fonction Lambda de manière synchrone

L'exemple `invoke` suivant invoque la fonction `my-function` de manière synchrone. L'`cli-binary-format`option est obligatoire si vous utilisez la version 2 de la AWS CLI. Pour plus d'informations, veuillez consulter les options de ligne de commande prises en charge par la CLI AWS dans le Guide de l'interface de ligne de commande AWS .

```
aws lambda invoke \
    --function-name my-function \
    --cli-binary-format raw-in-base64-out \
    --payload '{ "name": "Bob" }' \
```

```
    response.json
```

Sortie :

```
{
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
```

Pour plus d'informations, consultez la section Invoquer une fonction Lambda de manière synchrone dans le guide du développeur Lambda AWS .

Exemple 2 : pour invoquer une fonction Lambda de manière asynchrone

L'exemple `invoke` suivant invoque la fonction `my-function` de manière asynchrone. L'`cli-binary-format`option est obligatoire si vous utilisez la version 2 de la AWS CLI. Pour plus d'informations, veuillez consulter les options de ligne de commande prises en charge par la CLI AWS dans le Guide de l'interface de ligne de commande AWS .

```
aws lambda invoke \
    --function-name my-function \
    --invocation-type Event \
    --cli-binary-format raw-in-base64-out \
    --payload '{ "name": "Bob" }' \
    response.json
```

Sortie :

```
{
    "StatusCode": 202
}
```

Pour plus d'informations, consultez la section Invocation d'une fonction Lambda de manière asynchrone dans AWS le guide du développeur Lambda.

- Pour plus d'informations sur l'API, consultez Invoke dans la Référence des commandes AWS CLI .

`list-aliases`

L'exemple de code suivant montre comment utiliser`list-aliases`.

AWS CLI

Pour récupérer la liste des alias d'une fonction Lambda

L'`list-aliases`exemple suivant affiche la liste des alias de la fonction `my-function` Lambda.

```
aws lambda list-aliases \
    --function-name my-function
```

Sortie :

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:BETA",
            "RevisionId": "a410117f-ab16-494e-8035-7e204bb7933b",
            "FunctionVersion": "2",
            "Name": "BETA",
            "Description": "alias for beta version of function"
        },
        {
            "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:LIVE",
            "RevisionId": "21d40116-f8b1-40ba-9360-3ea284da1bb5",
            "FunctionVersion": "1",
            "Name": "LIVE",
            "Description": "alias for live version of function"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous ListAliasesà la section Référence des AWS CLI commandes.

`list-event-source-mappings`

L'exemple de code suivant montre comment utiliser`list-event-source-mappings`.

AWS CLI

Pour répertorier les mappages de sources d'événements pour une fonction

L'`list-event-source-mappings`exemple suivant affiche une liste des mappages de sources d'événements pour la fonction `my-function` Lambda.

```
aws lambda list-event-source-mappings \
    --function-name my-function
```

Sortie :

```
{
    "EventSourceMappings": [
        {
            "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "StateTransitionReason": "USER_INITIATED",
            "LastModified": 1569284520.333,
            "BatchSize": 5,
            "State": "Enabled",
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
            "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
        }
    ]
}
```

Pour plus d'informations, consultez la section Mappage des sources d'événements AWS Lambda dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous ListEventSourceMappingsà la section Référence des AWS CLI commandes.

`list-function-event-invoke-configs`

L'exemple de code suivant montre comment utiliser`list-function-event-invoke-configs`.

AWS CLI

Pour afficher la liste des configurations d'appel asynchrones

L'`list-function-event-invoke-configs`exemple suivant répertorie les configurations d'appel asynchrones pour la fonction spécifiée.

```
aws lambda list-function-event-invoke-configs \
    --function-name my-function
```

Sortie :

```
{
    "FunctionEventInvokeConfigs": [
        {
            "LastModified": 1577824406.719,
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:GREEN",
            "MaximumRetryAttempts": 2,
            "MaximumEventAgeInSeconds": 1800
        },
        {
            "LastModified": 1577824396.653,
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
            "MaximumRetryAttempts": 0,
            "MaximumEventAgeInSeconds": 3600
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListFunctionEventInvokeConfigsà la section Référence des AWS CLI commandes.

`list-functions`

L'exemple de code suivant montre comment utiliser`list-functions`.

AWS CLI

Pour récupérer la liste des fonctions Lambda

L'exemple `list-functions` suivant affiche une liste de toutes les fonctions pour l'utilisateur actuel.

```
aws lambda list-functions
```

Sortie :

```
{
    "Functions": [
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbL/RMr0fT/I=",
            "FunctionName": "helloworld",
            "MemorySize": 128,
            "RevisionId": "1718e831-badf-4253-9518-d0644210af7b",
            "CodeSize": 294,
            "FunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:helloworld",
            "Handler": "helloworld.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-
role-zgur6bf4",
            "Timeout": 3,
            "LastModified": "2023-09-23T18:32:33.857+0000",
            "Runtime": "nodejs18.x",
            "Description": ""
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
```

```
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2023-10-01T16:47:28.490+0000",
            "Runtime": "nodejs18.x",
            "Description": ""
        },
        {
            "Layers": [
                {
                    "CodeSize": 41784542,
                    "Arn": "arn:aws:lambda:us-west-2:420165488524:layer:AWSLambda-
Python37-SciPy1x:2"
                },
                {
                    "CodeSize": 4121,
                    "Arn": "arn:aws:lambda:us-
west-2:123456789012:layer:pythonLayer:1"
                }
            ],
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08mI=",
            "FunctionName": "my-python-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 128,
            "RevisionId": "80b4eabc-acf7-4ea8-919a-e874c213707d",
            "CodeSize": 299,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
python-function",
            "Handler": "lambda_function.lambda_handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/my-python-function-
role-z5g7dr6n",
            "Timeout": 3,
            "LastModified": "2023-10-01T19:40:41.643+0000",
            "Runtime": "python3.11",
            "Description": ""
```

```
        }
    ]
}
```

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous ListFunctionsà la section Référence des AWS CLI commandes.

`list-layer-versions`

L'exemple de code suivant montre comment utiliser`list-layer-versions`.

AWS CLI

Pour répertorier les versions d'une couche AWS Lambda

L'`list-layers-versions`exemple suivant affiche des informations sur les versions de la couche nomméemy`-layer`.

```
aws lambda list-layer-versions \
    --layer-name my-layer
```

Sortie :

```
{
    "Layers": [
        {
            "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
            "Version": 2,
            "Description": "My layer",
            "CreatedDate": "2023-11-15T00:37:46.592+0000",
            "CompatibleRuntimes": [
                "python3.10",
                "python3.11"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous ListLayerVersionsà la section Référence des AWS CLI commandes.

`list-layers`

L'exemple de code suivant montre comment utiliser`list-layers`.

AWS CLI

Pour répertorier les couches compatibles avec le temps d'exécution de votre fonction

L'`list-layers`exemple suivant affiche des informations sur les couches compatibles avec le moteur d'exécution Python 3.11.

```
aws lambda list-layers \
    --compatible-runtime python3.11
```

Sortie :

```
{
    "Layers": [
        {
            "LayerName": "my-layer",
            "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
            "LatestMatchingVersion": {
                "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
                "Version": 2,
                "Description": "My layer",
                "CreatedDate": "2023-11-15T00:37:46.592+0000",
                "CompatibleRuntimes": [
                    "python3.10",
                    "python3.11"
                ]
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous ListLayersà la section Référence des AWS CLI commandes.

`list-provisioned-concurrency-configs`

L'exemple de code suivant montre comment utiliser`list-provisioned-concurrency-configs`.

AWS CLI

Pour obtenir une liste des configurations de simultanéité allouée

L'exemple `list-provisioned-concurrency-configs` suivant répertorie les configurations de simultanéité allouée pour la fonction spécifiée.

```
aws lambda list-provisioned-concurrency-configs \
    --function-name my-function
```

Sortie :

```
{
    "ProvisionedConcurrencyConfigs": [
        {
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:GREEN",
            "RequestedProvisionedConcurrentExecutions": 100,
            "AvailableProvisionedConcurrentExecutions": 100,
            "AllocatedProvisionedConcurrentExecutions": 100,
            "Status": "READY",
            "LastModified": "2019-12-31T20:29:00+0000"
        },
        {
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
            "RequestedProvisionedConcurrentExecutions": 100,
            "AvailableProvisionedConcurrentExecutions": 100,
            "AllocatedProvisionedConcurrentExecutions": 100,
            "Status": "READY",
            "LastModified": "2019-12-31T20:28:49+0000"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListProvisionedConcurrencyConfigsà la section Référence des AWS CLI commandes.

`list-tags`

L'exemple de code suivant montre comment utiliser`list-tags`.

AWS CLI

Pour récupérer la liste des balises d'une fonction Lambda

L'exemple `list-tags` suivant affiche les balises associées à la fonction Lambda `my-function`.

```
aws lambda list-tags \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Sortie :

```
{
    "Tags": {
        "Category": "Web Tools",
        "Department": "Sales"
    }
}
```

Pour plus d'informations, consultez Étiquetage des fonctions Lambda dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous ListTagsà la section Référence des AWS CLI commandes.

`list-versions-by-function`

L'exemple de code suivant montre comment utiliser`list-versions-by-function`.

AWS CLI

Pour récupérer la liste des versions d'une fonction

L'exemple `list-versions-by-function` suivant affiche la liste des versions de la fonction Lambda `my-function`.

```
aws lambda list-versions-by-function \
    --function-name my-function
```

Sortie :

```
{
    "Versions": [
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:$LATEST",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-10-01T16:47:28.490+0000",
            "Runtime": "nodejs10.x",
            "Description": ""
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "1",
            "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",
            "FunctionName": "my-function",
            "VpcConfig": {
```

```
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "949c8914-012e-4795-998c-e467121951b1",
            "CodeSize": 304,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:1",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-09-26T20:28:40.438+0000",
            "Runtime": "nodejs10.x",
            "Description": "new version"
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "2",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "cd669f21-0f3d-4e1c-9566-948837f2e2ea",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:2",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-10-01T16:47:28.490+0000",
            "Runtime": "nodejs10.x",
            "Description": "newer version"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous ListVersionsByFunctionà la section Référence des AWS CLI commandes.

`publish-layer-version`

L'exemple de code suivant montre comment utiliser`publish-layer-version`.

AWS CLI

Pour créer une version de couche Lambda

L'`publish-layer-version`exemple suivant crée une nouvelle version de couche de bibliothèque Python. La commande récupère le contenu de la couche d'un fichier nommé `layer.zip` dans le compartiment S3 spécifié.

```
aws lambda publish-layer-version \
    --layer-name my-layer \
    --description "My Python layer" \
    --license-info "MIT" \
    --content S3Bucket=lambda-layers-us-west-2-123456789012,S3Key=layer.zip \
    --compatible-runtimes python3.10 python3.11
```

Sortie :

```
{
    "Content": {
        "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?
versionId=27iWyA73cCAYqyH...",
        "CodeSha256": "tv9jJO+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",
        "CodeSize": 169
    },
    "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer",
    "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1",
    "Description": "My Python layer",
    "CreatedDate": "2023-11-14T23:03:52.894+0000",
    "Version": 1,
```

```
    "LicenseInfo": "MIT",
    "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
    ]
}
```

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous PublishLayerVersionà la section Référence des AWS CLI commandes.

`publish-version`

L'exemple de code suivant montre comment utiliser`publish-version`.

AWS CLI

Pour publier une nouvelle version d'une fonction

L'exemple `publish-version` suivant publie une nouvelle version de la fonction Lambda `my-function`.

```
aws lambda publish-version \
    --function-name my-function
```

Sortie :

```
{
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbL/RMr0fT/I=",
    "FunctionName": "my-function",
    "CodeSize": 294,
    "RevisionId": "f31d3d39-cc63-4520-97d4-43cd44c94c20",
    "MemorySize": 128,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:3",
    "Version": "2",
    "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-
zgur6bf4",
    "Timeout": 3,
```

```
    "LastModified": "2019-09-23T18:32:33.857+0000",
    "Handler": "my-function.handler",
    "Runtime": "nodejs10.x",
    "Description": ""
}
```

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous PublishVersionà la section Référence des AWS CLI commandes.

`put-function-concurrency`

L'exemple de code suivant montre comment utiliser`put-function-concurrency`.

AWS CLI

Pour configurer une limite de simultanéité réservée pour une fonction

L'exemple `put-function-concurrency` suivant configure 100 exécutions simultanées réservées pour la fonction `my-function`.

```
aws lambda put-function-concurrency \
    --function-name  my-function  \
    --reserved-concurrent-executions 100
```

Sortie :

```
{
    "ReservedConcurrentExecutions": 100
}
```

Pour plus d'informations, consultez Réservation de simultanéité pour une fonction Lambda dans le Guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous PutFunctionConcurrencyà la section Référence des AWS CLI commandes.

`put-function-event-invoke-config`

L'exemple de code suivant montre comment utiliser`put-function-event-invoke-config`.

AWS CLI

Pour configurer la gestion des erreurs pour un appel asynchrone

L'`put-function-event-invoke-config`exemple suivant définit une durée maximale d'un événement d'une heure et désactive les nouvelles tentatives pour la fonction spécifiée.

```
aws lambda put-function-event-invoke-config \
    --function-name my-function \
    --maximum-event-age-in-seconds 3600 \
    --maximum-retry-attempts 0
```

Sortie :

```
{
    "LastModified": 1573686021.479,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:
$LATEST",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {}
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutFunctionEventInvokeConfigà la section Référence des AWS CLI commandes.

`put-provisioned-concurrency-config`

L'exemple de code suivant montre comment utiliser`put-provisioned-concurrency-config`.

AWS CLI

Pour allouer une simultanéité allouée

L'exemple `put-provisioned-concurrency-config` suivant alloue 100 simultanéités allouées à l'alias BLUE de la fonction spécifiée.

```
aws lambda put-provisioned-concurrency-config \
    --function-name my-function \
```

```
    --qualifier BLUE \
    --provisioned-concurrent-executions 100
```

Sortie :

```
{
    "Requested ProvisionedConcurrentExecutions": 100,
    "Allocated ProvisionedConcurrentExecutions": 0,
    "Status": "IN_PROGRESS",
    "LastModified": "2019-11-21T19:32:12+0000"
}
```

- Pour plus de détails sur l'API, reportez-vous PutProvisionedConcurrencyConfigà la section Référence des AWS CLI commandes.

`remove-layer-version-permission`

L'exemple de code suivant montre comment utiliser`remove-layer-version-permission`.

AWS CLI

Pour supprimer les autorisations de version de couche

L'`remove-layer-version-permission`exemple suivant supprime l'autorisation pour un compte de configurer une version de couche.

```
aws lambda remove-layer-version-permission \
    --layer-name my-layer \
    --statement-id xaccount \
    --version-number 1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section AWS Lambda Layers dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous RemoveLayerVersionPermissionà la section Référence des AWS CLI commandes.

`remove-permission`

L'exemple de code suivant montre comment utiliser`remove-permission`.

AWS CLI

Pour supprimer les autorisations d'une fonction Lambda existante

L'exemple `remove-permission` retire l'autorisation d'invoquer une fonction nommée `my-function`.

```
aws lambda remove-permission \
    --function-name my-function \
    --statement-id sns
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de politiques basées sur les ressources pour Lambda AWS dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous RemovePermissionà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une fonction Lambda existante

L'exemple `tag-resource` suivant ajoute une balise avec le nom de clé `DEPARTMENT` et une valeur de `Department A` à la fonction Lambda spécifiée.

```
aws lambda tag-resource \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --tags "DEPARTMENT=Department A"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Étiquetage des fonctions Lambda dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les balises d'une fonction Lambda existante

L'exemple `untag-resource` suivant supprime la balise dont la clé est `DEPARTMENT` de la fonction Lambda `my-function`.

```
aws lambda untag-resource \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --tag-keys DEPARTMENT
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Étiquetage des fonctions Lambda dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-alias`

L'exemple de code suivant montre comment utiliser`update-alias`.

AWS CLI

Pour mettre à jour un alias de fonction

L'exemple `update-alias` suivant met à jour l'alias de fonction Lambda nommé `LIVE` pour qu'il pointe vers la version 3 de la fonction Lambda `my-function`.

```
aws lambda update-alias \
    --function-name my-function \
    --function-version 3 \
    --name LIVE
```

Sortie :

```
{
```

```
    "FunctionVersion": "3",
    "Name": "LIVE",
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
    "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",
    "Description": "alias for live version of function"
}
```

Pour plus d'informations, consultez la section Configuration des alias de fonction AWS Lambda dans le guide du développeur Lambda AWS .

- Pour plus de détails sur l'API, reportez-vous UpdateAliasà la section Référence des AWS CLI commandes.

`update-event-source-mapping`

L'exemple de code suivant montre comment utiliser`update-event-source-mapping`.

AWS CLI

Pour mettre à jour le mappage entre une source d'événements et une fonction AWS Lambda

L'`update-event-source-mapping`exemple suivant met à jour la taille du lot à 8 dans le mappage spécifié.

```
aws lambda update-event-source-mapping \
    --uuid  "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --batch-size 8
```

Sortie :

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569284520.333,
    "BatchSize": 8,
    "State": "Updating",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

Pour plus d'informations, consultez la section Mappage des sources d'événements AWS Lambda dans le guide du développeur AWS Lambda.

- Pour plus de détails sur l'API, reportez-vous UpdateEventSourceMappingà la section Référence des AWS CLI commandes.

`update-function-code`

L'exemple de code suivant montre comment utiliser`update-function-code`.

AWS CLI

Pour mettre à jour le code d'une fonction Lambda

L'exemple `update-function-code` suivant remplace le code de la version non publiée ($LATEST) de la fonction `my-function` par le contenu du fichier zip spécifié.

```
aws lambda update-function-code \
    --function-name  my-function \
    --zip-file fileb://my-function.zip
```

Sortie :

```
{
    "FunctionName": "my-function",
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
    "MemorySize": 256,
    "Version": "$LATEST",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler"
```

```
}
```

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous UpdateFunctionCodeà la section Référence des AWS CLI commandes.

`update-function-configuration`

L'exemple de code suivant montre comment utiliser`update-function-configuration`.

AWS CLI

Pour modifier la configuration d'une fonction

L'exemple `update-function-configuration` suivant modifie la taille de la mémoire à 256 Mo pour la version non publiée ($LATEST) de la fonction `my-function`.

```
aws lambda update-function-configuration \
    --function-name  my-function \
    --memory-size 256
```

Sortie :

```
{
    "FunctionName": "my-function",
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
    "MemorySize": 256,
    "Version": "$LATEST",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
```

```
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler"
}
```

Pour plus d'informations, consultez Configuration des options de fonction Lambda AWS dans le Guide du développeur AWS .

- Pour plus de détails sur l'API, reportez-vous UpdateFunctionConfigurationà la section Référence des AWS CLI commandes.

`update-function-event-invoke-config`

L'exemple de code suivant montre comment utiliser`update-function-event-invoke-config`.

AWS CLI

Pour mettre à jour une configuration d'appel asynchrone

L'`update-function-event-invoke-config`exemple suivant ajoute une destination en cas d'échec à la configuration d'appel asynchrone existante pour la fonction spécifiée.

```
aws lambda update-function-event-invoke-config \
    --function-name my-function \
    --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-
east-2:123456789012:destination"}}'
```

Sortie :

```
{
    "LastModified": 1573687896.493,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:
$LATEST",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {
            "Destination": "arn:aws:sqs:us-east-2:123456789012:destination"
        }
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateFunctionEventInvokeConfigà la section Référence des AWS CLI commandes.

Exemples de License Manager utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with License Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-license-configuration`

L'exemple de code suivant montre comment utiliser`create-license-configuration`.

AWS CLI

Exemple 1 : pour créer une configuration de licence

L'`create-license-configuration`exemple suivant crée une configuration de licence avec une limite stricte de 10 cœurs.

```
aws license-manager create-license-configuration --name my-license-configuration \
    --license-counting-type Core \
    --license-count 10 \
    --license-count-hard-limit
```

Sortie :

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"
}
```

Exemple 2 : pour créer une configuration de licence

L'`create-license-configuration`exemple suivant crée une configuration de licence avec une limite souple de 100 CPUs v. Il utilise une règle pour activer l'optimisation des vCPU.

```
aws license-manager create-license-configuration --name my-license-configuration
    --license-counting-type vCPU \
    --license-count 100 \
    --license-rules "#honorVcpuOptimization=true"
```

Sortie :

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLicenseConfigurationà la section Référence des AWS CLI commandes.

`delete-license-configuration`

L'exemple de code suivant montre comment utiliser`delete-license-configuration`.

AWS CLI

Pour supprimer une configuration de licence

L'`delete-license-configuration`exemple suivant supprime la configuration de licence spécifiée.

```
aws license-manager delete-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteLicenseConfigurationà la section Référence des AWS CLI commandes.

`get-license-configuration`

L'exemple de code suivant montre comment utiliser`get-license-configuration`.

AWS CLI

Pour obtenir des informations de configuration de licence

L'`get-license-configuration`exemple suivant affiche les détails de la configuration de licence spécifiée.

```
aws license-manager get-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Sortie :

```
{
    "LicenseConfigurationId": "lic-38b658717b87478aaa7c00883EXAMPLE",
    "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE",
    "Name": "my-license-configuration",
    "LicenseCountingType": "vCPU",
    "LicenseRules": [],
    "LicenseCountHardLimit": false,
    "ConsumedLicenses": 0,
    "Status": "AVAILABLE",
    "OwnerAccountId": "123456789012",
    "ConsumedLicenseSummaryList": [
        {
            "ResourceType": "EC2_INSTANCE",
            "ConsumedLicenses": 0
        },
        {
            "ResourceType": "EC2_HOST",
            "ConsumedLicenses": 0
        },
        {
            "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
```

```
            "ConsumedLicenses": 0
        }
    ],
    "ManagedResourceSummaryList": [
        {
            "ResourceType": "EC2_INSTANCE",
            "AssociationCount": 0
        },
        {
            "ResourceType": "EC2_HOST",
            "AssociationCount": 0
        },
        {
            "ResourceType": "EC2_AMI",
            "AssociationCount": 2
        },
        {
            "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
            "AssociationCount": 0
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLicenseConfigurationà la section Référence des AWS CLI commandes.

`get-service-settings`

L'exemple de code suivant montre comment utiliser`get-service-settings`.

AWS CLI

Pour obtenir les paramètres du License Manager

L'`get-service-settings`exemple suivant affiche les paramètres de service pour License Manager dans la région actuelle.

```
aws license-manager get-service-settings
```

Voici un exemple de sortie si la découverte de ressources entre comptes est désactivée.

```
{
```

```
    "OrganizationConfiguration": {
        "EnableIntegration": false
    },
    "EnableCrossAccountsDiscovery": false
}
```

Voici un exemple de sortie si la découverte de ressources entre comptes est activée.

```
{
    "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
    "OrganizationConfiguration": {
        "EnableIntegration": true
    },
    "EnableCrossAccountsDiscovery": true
}
```

- Pour plus de détails sur l'API, reportez-vous GetServiceSettingsà la section Référence des AWS CLI commandes.

`list-associations-for-license-configuration`

L'exemple de code suivant montre comment utiliser`list-associations-for-license-configuration`.

AWS CLI

Pour obtenir des associations pour une configuration de licence

L'`list-associations-for-license-configuration`exemple suivant affiche des informations détaillées sur les associations de la configuration de licence spécifiée.

```
aws license-manager list-associations-for-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Sortie :

```
{
    "LicenseConfigurationAssociations": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0",
```

```
            "ResourceType": "EC2_AMI",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1568825118.617
        },
        {
            "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-0abcdef1234567890",
            "ResourceType": "EC2_AMI",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1568825118.946
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAssociationsForLicenseConfigurationà la section Référence des AWS CLI commandes.

`list-license-configurations`

L'exemple de code suivant montre comment utiliser`list-license-configurations`.

AWS CLI

Exemple 1 : pour répertorier toutes vos configurations de licence

L'`list-license-configurations`exemple suivant répertorie toutes vos configurations de licence.

```
aws license-manager list-license-configurations
```

Sortie :

```
{
    "LicenseConfigurations": [
        {
            "LicenseConfigurationId": "lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
            "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
            "Name": "my-license-configuration",
            "LicenseCountingType": "Core",
            "LicenseRules": [],
            "LicenseCount": 10,
            "LicenseCountHardLimit": true,
```

```
            "ConsumedLicenses": 0,
            "Status": "AVAILABLE",
            "OwnerAccountId": "123456789012",
            "ConsumedLicenseSummaryList": [
                {
                    "ResourceType": "EC2_INSTANCE",
                    "ConsumedLicenses": 0
                },
                {
                    "ResourceType": "EC2_HOST",
                    "ConsumedLicenses": 0
                },
                {
                    "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
                    "ConsumedLicenses": 0
                }
            ],
            "ManagedResourceSummaryList": [
                {
                    "ResourceType": "EC2_INSTANCE",
                    "AssociationCount": 0
                },
                {
                    "ResourceType": "EC2_HOST",
                    "AssociationCount": 0
                },
                {
                    "ResourceType": "EC2_AMI",
                    "AssociationCount": 0
                },
                {
                    "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
                    "AssociationCount": 0
                }
            ]
        },
        {
            ...
        }
    ]
}
```

Exemple 2 : pour répertorier une configuration de licence spécifique

L'`list-license-configurations`exemple suivant répertorie uniquement la configuration de licence spécifiée.

```
aws license-manager list-license-configurations \
    --license-configuration-arns arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous ListLicenseConfigurationsà la section Référence des AWS CLI commandes.

`list-license-specifications-for-resource`

L'exemple de code suivant montre comment utiliser`list-license-specifications-for-resource`.

AWS CLI

Pour répertorier les configurations de licence pour une ressource

L'`list-license-specifications-for-resource`exemple suivant répertorie les configurations de licence associées à l'Amazon Machine Image (AMI) spécifiée.

```
aws license-manager list-license-specifications-for-resource \
    --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0
```

Sortie :

```
{
    "LicenseConfigurationArn": "arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"
}
```

- Pour plus de détails sur l'API, reportez-vous ListLicenseSpecificationsForResourceà la section Référence des AWS CLI commandes.

`list-resource-inventory`

L'exemple de code suivant montre comment utiliser`list-resource-inventory`.

AWS CLI

Pour répertorier les ressources dans l'inventaire des ressources

L'`list-resource-inventory`exemple suivant répertorie les ressources gérées à l'aide de l'inventaire de Systems Manager.

```
aws license-manager list-resource-inventory
```

Sortie :

```
{
    "ResourceInventoryList": [
        {
            "Platform": "Red Hat Enterprise Linux Server",
            "ResourceType": "EC2Instance",
            "PlatformVersion": "7.4",
            "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-05d3cdfb05bd36376",
            "ResourceId": "i-05d3cdfb05bd36376",
            "ResourceOwningAccountId": "1234567890129"
        },
        {
            "Platform": "Amazon Linux",
            "ResourceType": "EC2Instance",
            "PlatformVersion": "2",
            "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0b1d036cfd4594808",
            "ResourceId": "i-0b1d036cfd4594808",
            "ResourceOwningAccountId": "1234567890129"
        },
        {
            "Platform": "Microsoft Windows Server 2019 Datacenter",
            "ResourceType": "EC2Instance",
            "PlatformVersion": "10.0.17763",
            "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0cdb3b54a2a8246ad",
            "ResourceId": "i-0cdb3b54a2a8246ad",
            "ResourceOwningAccountId": "1234567890129"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceInventoryà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une configuration de licence

L'`list-tags-for-resource`exemple suivant répertorie les balises correspondant à la configuration de licence spécifiée.

```
aws license-manager list-tags-for-resource \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "project",
            "Value": "lima"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-usage-for-license-configuration`

L'exemple de code suivant montre comment utiliser`list-usage-for-license-configuration`.

AWS CLI

Pour répertorier les licences utilisées pour une configuration de licence

L'`list-usage-for-license-configuration`exemple suivant répertorie les informations relatives aux ressources utilisant des licences pour la configuration de licence spécifiée. Par exemple, si le type de licence est vCPU, toutes les instances consomment une licence par vCPU.

```
aws license-manager list-usage-for-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Sortie :

```
{
    "LicenseConfigurationUsageList": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-04a636d18e83cfacb",
            "ResourceType": "EC2_INSTANCE",
            "ResourceStatus": "running",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1570892850.519,
            "ConsumedLicenses": 2
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListUsageForLicenseConfigurationà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter un tag, une configuration de licence

L'`tag-resource`exemple suivant ajoute la balise spécifiée (nom de clé et valeur) à la configuration de licence spécifiée.

```
aws license-manager tag-resource \
    --tags Key=project,Value=lima \
```

```
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une configuration de licence

L'`untag-resource`exemple suivant supprime la balise spécifiée (nom de clé et ressource) de la configuration de licence spécifiée.

```
aws license-manager untag-resource \
    --tag-keys project \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-license-configuration`

L'exemple de code suivant montre comment utiliser`update-license-configuration`.

AWS CLI

Pour mettre à jour la configuration d'une licence

L'`update-license-configuration`exemple suivant met à jour la configuration de licence spécifiée pour supprimer la limite stricte.

```
aws license-manager update-license-configuration \
```

```
    --no-license-count-hard-limit \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Cette commande ne produit aucun résultat.

L'`update-license-configuration`exemple suivant met à jour la configuration de licence spécifiée pour changer son statut en`DISABLED`.

```
aws license-manager update-license-configuration \
    --license-configuration-status DISABLED
    --license-configuration-arn arn:aws:license-manager:us-
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateLicenseConfigurationà la section Référence des AWS CLI commandes.

`update-license-specifications-for-resource`

L'exemple de code suivant montre comment utiliser`update-license-specifications-for-resource`.

AWS CLI

Pour mettre à jour les configurations de licence pour une ressource

L'`update-license-specifications-for-resource`exemple suivant remplace la configuration de licence associée à l'Amazon Machine Image (AMI) spécifiée en supprimant une configuration de licence et en en ajoutant une autre.

```
aws license-manager update-license-specifications-for-resource \
    --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0 \
    --remove-license-specifications LicenseConfigurationArn=arn:aws:license-
manager:us-west-2:123456789012:license-
configuration:lic-38b658717b87478aaa7c00883EXAMPLE \
    --add-license-specifications LicenseConfigurationArn=arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-42b6deb06e5399a980d555927EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateLicenseSpecificationsForResourceà la section Référence des AWS CLI commandes.

`update-service-settings`

L'exemple de code suivant montre comment utiliser`update-service-settings`.

AWS CLI

Pour mettre à jour les paramètres du License Manager

L'`update-service-settings`exemple suivant permet la découverte de ressources entre comptes pour License Manager dans la AWS région actuelle. Le compartiment Amazon S3 est la synchronisation des données de ressources requise pour l'inventaire de Systems Manager.

```
aws license-manager update-service-settings \
    --organization-configuration EnableIntegration=true \
    --enable-cross-accounts-discovery \
    --s3-bucket-arn arn:aws:s3:::aws-license-manager-service-abcd1234EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceSettingsà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Lightsail AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Lightsail.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`allocate-static-ip`

L'exemple de code suivant montre comment utiliser`allocate-static-ip`.

AWS CLI

Pour créer une adresse IP statique

L'`allocate-static-ip`exemple suivant crée l'adresse IP statique spécifiée, qui peut être attachée à une instance.

```
aws lightsail allocate-static-ip \
    --static-ip-name StaticIp-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "b5d06d13-2f19-4683-889f-dEXAMPLEed79",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571071325.076,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "AllocateStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571071325.274
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AllocateStaticIpà la section Référence des AWS CLI commandes.

`attach-disk`

L'exemple de code suivant montre comment utiliser`attach-disk`.

AWS CLI

Pour associer un disque de stockage par blocs à une instance

L'`attach-disk`exemple suivant attache un disque `Disk-1` à une instance `WordPress_Multisite-1` avec le chemin de disque de `/dev/xvdf`

```
aws lightsail attach-disk \
    --disk-name Disk-1 \
    --disk-path /dev/xvdf \
    --instance-name WordPress_Multisite-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "10a08267-19ce-43be-b913-6EXAMPLE7e80",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1571071465.472,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress_Multisite-1",
            "operationType": "AttachDisk",
            "status": "Started",
            "statusChangedAt": 1571071465.472
        },
        {
            "id": "2912c477-5295-4539-88c9-bEXAMPLEd1f0",
            "resourceName": "WordPress_Multisite-1",
            "resourceType": "Instance",
            "createdAt": 1571071465.474,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
```

```
            },
            "isTerminal": false,
            "operationDetails": "Disk-1",
            "operationType": "AttachDisk",
            "status": "Started",
            "statusChangedAt": 1571071465.474
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AttachDiskà la section Référence des AWS CLI commandes.

`attach-instances-to-load-balancer`

L'exemple de code suivant montre comment utiliser`attach-instances-to-load-balancer`.

AWS CLI

Pour associer des instances à un équilibreur de charge

L'`attach-instances-to-load-balancer`exemple suivant attache des instances `MEAN-1``MEAN-2`, et `MEAN-3` à l'équilibreur `LoadBalancer-1` de charge.

```
aws lightsail attach-instances-to-load-balancer \
    --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \
    --load-balancer-name LoadBalancer-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "8055d19d-abb2-40b9-b527-1EXAMPLE3c7b",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.892,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
```

```
            "operationDetails": "MEAN-2",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.892
        },
        {
            "id": "c35048eb-8538-456a-a118-0EXAMPLEfb73",
            "resourceName": "MEAN-2",
            "resourceType": "Instance",
            "createdAt": 1571071699.887,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.887
        },
        {
            "id": "910d09e0-adc5-4372-bc2e-0EXAMPLEd891",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.882,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-3",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.882
        },
        {
            "id": "178b18ac-43e8-478c-9bed-1EXAMPLE4755",
            "resourceName": "MEAN-3",
            "resourceType": "Instance",
            "createdAt": 1571071699.901,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.901
        },
        {
            "id": "fb62536d-2a98-4190-a6fc-4EXAMPLE7470",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.885,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.885
        },
        {
            "id": "787dac0d-f98d-46c3-8571-3EXAMPLE5a85",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571071699.901,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.901
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous AttachInstancesToLoadBalancerà la section Référence des AWS CLI commandes.

attach-load-balancer-tls-certificate

L'exemple de code suivant montre comment utiliserattach-load-balancer-tls-certificate.

AWS CLI

Pour associer un certificat TLS à un équilibreur de charge

L'attach-load-balancer-tls-certificateexemple suivant attache le certificat TLS de l'équilibreur de charge Certificate2 à l'équilibreur de charge. LoadBalancer-1

```
aws lightsail attach-load-balancer-tls-certificate \
    --certificate-name Certificate2 \
    --load-balancer-name LoadBalancer-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "cf1ad6e3-3cbb-4b8a-a7f2-3EXAMPLEa118",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571072255.416,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "Certificate2",
            "operationType": "AttachLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1571072255.416
        },
        {
            "id": "dae1bcfb-d531-4c06-b4ea-bEXAMPLEc04e",
            "resourceName": "Certificate2",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1571072255.416,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "isTerminal": true,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1571072255.416
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AttachLoadBalancerTlsCertificateà la section Référence des AWS CLI commandes.

`attach-static-ip`

L'exemple de code suivant montre comment utiliser`attach-static-ip`.

AWS CLI

Pour associer une adresse IP statique à une instance

L'`attach-static-ip`exemple suivant attache une adresse IP statique `StaticIp-1` à l'instance`MEAN-1`.

```
aws lightsail attach-static-ip \
    --static-ip-name StaticIp-1 \
    --instance-name MEAN-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "45e6fa13-4808-4b8d-9292-bEXAMPLE20b2",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571072569.375,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
```

```
            "operationDetails": "MEAN-1",
            "operationType": "AttachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571072569.375
        },
        {
            "id": "9ee09a17-863c-4e51-8a6d-3EXAMPLE5475",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571072569.376,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "StaticIp-1",
            "operationType": "AttachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571072569.376
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AttachStaticIpà la section Référence des AWS CLI commandes.

`close-instance-public-ports`

L'exemple de code suivant montre comment utiliser`close-instance-public-ports`.

AWS CLI

Pour fermer les ports de pare-feu d'une instance

L'`close-instance-public-ports`exemple suivant ferme le port TCP 22 sur l'instance`MEAN-2`.

```
aws lightsail close-instance-public-ports \
    --instance-name MEAN-2 \
    --port-info fromPort=22,protocol=TCP,toPort=22
```

Sortie :

```
{
    "operation": {
        "id": "4f328636-1c96-4649-ae6d-1EXAMPLEf446",
        "resourceName": "MEAN-2",
        "resourceType": "Instance",
        "createdAt": 1571072845.737,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "22/tcp",
        "operationType": "CloseInstancePublicPorts",
        "status": "Succeeded",
        "statusChangedAt": 1571072845.737
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CloseInstancePublicPortsà la section Référence des AWS CLI commandes.

`copy-snapshot`

L'exemple de code suivant montre comment utiliser`copy-snapshot`.

AWS CLI

Exemple 1 : pour copier un instantané dans la même AWS région

L'`copy-snapshot`exemple suivant copie un instantané d'instance `MEAN-1-1571075291` en tant qu'instantané d'instance `MEAN-1-Copy` dans la même AWS région`us-west-2`.

```
aws lightsail copy-snapshot \
    --source-snapshot-name MEAN-1-1571075291 \
    --target-snapshot-name MEAN-1-Copy \
    --source-region us-west-2
```

Sortie :

```
{
    "operations": [
        {
```

```
            "id": "ced16fc1-f401-4556-8d82-1EXAMPLEb982",
            "resourceName": "MEAN-1-Copy",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571075581.498,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:MEAN-1-1571075291",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571075581.498
        }
    ]
}
```

Pour plus d'informations, consultez Copier des instantanés d'une AWS région à une autre dans Amazon Lightsail dans le Guide de développement de Lightsail.

Exemple 2 : pour copier un instantané d'une AWS région à une autre

L'`copy-snapshot`exemple suivant copie un instantané d'instance `MEAN-1-1571075291` en tant qu'instantané `MEAN-1-1571075291-Copy` d'instance de AWS Region `us-west-2` versus-`east-1`.

```
aws lightsail copy-snapshot \
    --source-snapshot-name MEAN-1-1571075291 \
    --target-snapshot-name MEAN-1-1571075291-Copy \
    --source-region us-west-2 \
    --region us-east-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "91116b79-119c-4451-b44a-dEXAMPLEd97b",
            "resourceName": "MEAN-1-1571075291-Copy",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571075695.069,
            "location": {
                "availabilityZone": "all",
```

```
                "regionName": "us-east-1"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:MEAN-1-1571075291",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571075695.069
        }
    ]
}
```

Pour plus d'informations, consultez Copier des instantanés d'une AWS région à une autre dans Amazon Lightsail dans le Guide de développement de Lightsail.

Exemple 3 : pour copier un instantané automatique dans la même AWS région

L'`copy-snapshot`exemple suivant copie un instantané automatique `2019-10-14` de l'instance `WordPress-1` sous la forme d'un instantané manuel `WordPress-1-10142019` dans la AWS région`us-west-2`.

```
aws lightsail copy-snapshot \
    --source-resource-name WordPress-1 \
    --restore-date 2019-10-14 \
    --target-snapshot-name WordPress-1-10142019 \
    --source-region us-west-2
```

Sortie :

```
{
    "operations": [
        {
            "id": "be3e6754-cd1d-48e6-ad9f-2EXAMPLE1805",
            "resourceName": "WordPress-1-10142019",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571082412.311,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:WordPress-1",
            "operationType": "CopySnapshot",
            "status": "Started",
```

```
            "statusChangedAt": 1571082412.311
        }
    ]
}
```

Pour plus d'informations, consultez la section Conservation des instantanés automatiques des instances ou des disques dans Amazon Lightsail dans le Guide de développement de Lightsail.

Exemple 4 : pour copier un instantané automatique d'une AWS région à une autre

L'`copy-snapshot`exemple suivant copie un instantané automatique `2019-10-14` de l'instance `WordPress-1` sous la forme d'un instantané manuel `WordPress-1-10142019` de la AWS région `us-west-2` vers`us-east-1`.

```
aws lightsail copy-snapshot \
    --source-resource-name WordPress-1 \
    --restore-date 2019-10-14 \
    --target-snapshot-name WordPress-1-10142019 \
    --source-region us-west-2 \
    --region us-east-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "dffa128b-0b07-476e-b390-bEXAMPLE3775",
            "resourceName": "WordPress-1-10142019",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571082493.422,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-1"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:WordPress-1",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571082493.422
        }
    ]
}
```

Pour plus d'informations, consultez la section Conservation des instantanés automatiques des instances ou des disques dans Amazon Lightsail dans le Guide de développement de Lightsail.

- Pour plus de détails sur l'API, reportez-vous CopySnapshotà la section Référence des AWS CLI commandes.

`create-disk-from-snapshot`

L'exemple de code suivant montre comment utiliser`create-disk-from-snapshot`.

AWS CLI

Pour créer un, créez un disque à partir d'un instantané de disque

L'`create-disk-from-snapshot`exemple suivant crée un disque de stockage par blocs nommé `Disk-2` à partir de l'instantané du disque de stockage par blocs spécifié. Le disque est créé dans la AWS région et la zone de disponibilité spécifiées, avec 32 Go d'espace de stockage.

```
aws lightsail create-disk-from-snapshot \
    --disk-name Disk-2 \
    --disk-snapshot-name Disk-1-1566839161 \
    --availability-zone us-west-2a \
    --size-in-gb 32
```

Sortie :

```
{
    "operations": [
        {
            "id": "d42b605d-5ef1-4b4a-8791-7a3e8b66b5e7",
            "resourceName": "Disk-2",
            "resourceType": "Disk",
            "createdAt": 1569624941.471,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateDiskFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1569624941.791
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Création d'un disque de stockage par blocs à partir d'un instantané dans Amazon Lightsail dans le Guide du développeur de Lightsail.

- Pour plus de détails sur l'API, reportez-vous CreateDiskFromSnapshotà la section Référence des AWS CLI commandes.

`create-disk-snapshot`

L'exemple de code suivant montre comment utiliser`create-disk-snapshot`.

AWS CLI

Exemple 1 : pour créer un instantané d'un disque

L'`create-disk-snapshot`exemple suivant crée un instantané nommé `DiskSnapshot-1` du disque de stockage par blocs spécifié.

```
aws lightsail create-disk-snapshot \
    --disk-name Disk-1 \
    --disk-snapshot-name DiskSnapshot-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "fa74c6d2-03a3-4f42-a7c7-792f124d534b",
            "resourceName": "DiskSnapshot-1",
            "resourceType": "DiskSnapshot",
            "createdAt": 1569625129.739,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Disk-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625129.739
        },
        {
```

```
            "id": "920a25df-185c-4528-87cd-7b85f5488c06",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569625129.739,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "DiskSnapshot-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625129.739
        }
    ]
}
```

Exemple 2 : pour créer un instantané du disque système d'une instance

L'`create-disk-snapshot`exemple suivant crée un instantané du disque système de l'instance spécifiée.

```
aws lightsail create-disk-snapshot \
    --instance-name WordPress-1 \
    --disk-snapshot-name SystemDiskSnapshot-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "f508cf1c-6597-42a6-a4c3-4aebd75af0d9",
            "resourceName": "SystemDiskSnapshot-1",
            "resourceType": "DiskSnapshot",
            "createdAt": 1569625294.685,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
```

```
            "statusChangedAt": 1569625294.685
        },
        {
            "id": "0bb9f712-da3b-4d99-b508-3bf871d989e5",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569625294.685,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "SystemDiskSnapshot-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625294.685
        }
    ]
}
```

Pour plus d'informations, consultez les sections Instantanés dans Amazon Lightsail et Création d'un instantané d'un volume racine d'instance dans Amazon Lightsail dans le Guide du développeur de Lightsail.

- Pour plus de détails sur l'API, reportez-vous CreateDiskSnapshotà la section Référence des AWS CLI commandes.

`create-disk`

L'exemple de code suivant montre comment utiliser`create-disk`.

AWS CLI

Pour créer un disque de stockage par blocs

L'`create-disk`exemple suivant crée un disque de stockage par blocs `Disk-1` dans la AWS région et la zone de disponibilité spécifiées, avec 32 Go d'espace de stockage.

```
aws lightsail create-disk \
    --disk-name Disk-1 \
    --availability-zone us-west-2a \
    --size-in-gb 32
```

Sortie :

```
{
    "operations": [
        {
            "id": "1c85e2ec-86ba-4697-b936-77f4d3dc013a",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569449220.36,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateDisk",
            "status": "Started",
            "statusChangedAt": 1569449220.588
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDiskà la section Référence des AWS CLI commandes.

`create-domain-entry`

L'exemple de code suivant montre comment utiliser`create-domain-entry`.

AWS CLI

Pour créer une entrée de domaine (enregistrement DNS)

L'`create-domain-entry`exemple suivant crée un enregistrement DNS (A) pour le sommet du domaine spécifié qui pointe vers l'adresse IP d'une instance.

Remarque : les opérations d'API liées au domaine de Lightsail ne sont disponibles que dans la région. `us-east-1` Si votre profil CLI est configuré pour utiliser une région différente, vous devez inclure le `--region us-east-1` paramètre, sinon la commande échoue.

```
aws lightsail create-domain-entry \
    --region us-east-1 \
    --domain-name example.com \
```

```
    --domain-entry name=example.com,type=A,target=192.0.2.0
```

Sortie :

```
{
    "operation": {
        "id": "5be4494d-56f4-41fc-8730-693dcd0ef9e2",
        "resourceName": "example.com",
        "resourceType": "Domain",
        "createdAt": 1569865296.519,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "CreateDomainEntry",
        "status": "Succeeded",
        "statusChangedAt": 1569865296.519
    }
}
```

Pour plus d'informations, consultez les sections DNS dans Amazon Lightsail et Création d'une zone DNS pour gérer les enregistrements DNS de votre domaine dans Amazon Lightsail dans le guide du développeur de Lightsail.

- Pour plus de détails sur l'API, reportez-vous CreateDomainEntryà la section Référence des AWS CLI commandes.

`create-domain`

L'exemple de code suivant montre comment utiliser`create-domain`.

AWS CLI

Pour créer un domaine (zone DNS)

L'`create-domain`exemple suivant crée une zone DNS pour le domaine spécifié.

Remarque : les opérations d'API liées au domaine de Lightsail ne sont disponibles que dans la région. `us-east-1` Si votre profil CLI est configuré pour utiliser une région différente, vous devez inclure le `--region us-east-1` paramètre, sinon la commande échoue.

```
aws lightsail create-domain \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "operation": {
        "id": "64e522c8-9ae1-4c05-9b65-3f237324dc34",
        "resourceName": "example.com",
        "resourceType": "Domain",
        "createdAt": 1569864291.92,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "CreateDomain",
        "status": "Succeeded",
        "statusChangedAt": 1569864292.109
    }
}
```

Pour plus d'informations, consultez les sections DNS dans Amazon Lightsail et Création d'une zone DNS pour gérer les enregistrements DNS de votre domaine dans Amazon Lightsail dans le guide du développeur de Lightsail.

- Pour plus de détails sur l'API, reportez-vous CreateDomainà la section Référence des AWS CLI commandes.

`create-instance-snapshot`

L'exemple de code suivant montre comment utiliser`create-instance-snapshot`.

AWS CLI

Pour créer un instantané d'une instance

L'`create-instance-snapshot`exemple suivant crée un instantané à partir de l'instance spécifiée.

```
aws lightsail create-instance-snapshot \
```

```
    --instance-name WordPress-1 \
    --instance-snapshot-name WordPress-Snapshot-1
```

Sortie :


```
{
    "operations": [
        {
            "id": "4c3db559-9dd0-41e7-89c0-2cb88c19786f",
            "resourceName": "WordPress-Snapshot-1",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1569866438.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-1",
            "operationType": "CreateInstanceSnapshot",
            "status": "Started",
            "statusChangedAt": 1569866438.48
        },
        {
            "id": "c04fdc45-2981-488c-88b5-d6d2fd759a6a",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569866438.48,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-Snapshot-1",
            "operationType": "CreateInstanceSnapshot",
            "status": "Started",
            "statusChangedAt": 1569866438.48
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous CreateInstanceSnapshotà la section Référence des AWS CLI commandes.

`create-instances-from-snapshot`

L'exemple de code suivant montre comment utiliser`create-instances-from-snapshot`.

AWS CLI

Pour créer une instance à partir d'un instantané

L'`create-instances-from-snapshot`exemple suivant crée une instance à partir de l'instantané d'instance spécifié, dans la AWS région et la zone de disponibilité spécifiées, en utilisant le bundle de 12 USD.

Remarque : Le bundle que vous spécifiez doit avoir des spécifications égales ou supérieures à celles du bundle de l'instance source d'origine utilisée pour créer le cliché.

```
aws lightsail create-instances-from-snapshot \
    --instance-snapshot-name WordPress-1-1569866208 \
    --instance-names WordPress-2 \
    --availability-zone us-west-2a \
    --bundle-id small_3_0
```

Sortie :

```
{
    "operations": [
        {
            "id": "003f8271-b711-464d-b9b8-7f3806cb496e",
            "resourceName": "WordPress-2",
            "resourceType": "Instance",
            "createdAt": 1569865914.908,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstancesFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1569865914.908
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateInstancesFromSnapshotà la section Référence des AWS CLI commandes.

`create-instances`

L'exemple de code suivant montre comment utiliser`create-instances`.

AWS CLI

Exemple 1 : pour créer une instance unique

L'`create-instances`exemple suivant crée une instance dans la AWS région et la zone de disponibilité spécifiées, à l'aide du WordPress plan et du bundle de 5 USD.

```
aws lightsail create-instances \
    --instance-names Instance-1 \
    --availability-zone us-west-2a \
    --blueprint-id wordpress \
    --bundle-id nano_3_0
```

Sortie :

```
{
    "operations": [
        {
            "id": "9a77158f-7be3-4d6d-8054-cf5ae2b720cc",
            "resourceName": "Instance-1",
            "resourceType": "Instance",
            "createdAt": 1569447986.061,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569447986.061
        }
    ]
}
```

Exemple 2 : pour créer plusieurs instances à la fois

L'`create-instances`exemple suivant crée trois instances dans la AWS région et la zone de disponibilité spécifiées, à l'aide du WordPress plan et du bundle de 5 USD.

```
aws lightsail create-instances \
    --instance-names {"Instance1","Instance2","Instance3"} \
    --availability-zone us-west-2a \
    --blueprint-id wordpress \
    --bundle-id nano_3_0
```

Sortie :

```
{
    "operations": [
        {
            "id": "5492f015-9d2e-48c6-8eea-b516840e6903",
            "resourceName": "Instance1",
            "resourceType": "Instance",
            "createdAt": 1569448780.054,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569448780.054
        },
        {
            "id": "c58b5f46-2676-44c8-b95c-3ad375898515",
            "resourceName": "Instance2",
            "resourceType": "Instance",
            "createdAt": 1569448780.054,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569448780.054
        },
        {
            "id": "a5ad8006-9bee-4499-9eb7-75e42e6f5882",
```

```
            "resourceName": "Instance3",
            "resourceType": "Instance",
            "createdAt": 1569448780.054,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569448780.054
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateInstancesà la section Référence des AWS CLI commandes.

`create-key-pair`

L'exemple de code suivant montre comment utiliser`create-key-pair`.

AWS CLI

Création d'une paire de clés

L'`create-key-pair`exemple suivant crée une paire de clés que vous pouvez utiliser pour vous authentifier et vous connecter à une instance.

```
aws lightsail create-key-pair \
    --key-pair-name MyPersonalKeyPair
```

La sortie fournit la valeur de clé privée base64 que vous pouvez utiliser pour vous authentifier auprès des instances qui utilisent la paire de clés créée. Remarque : Copiez et collez la valeur de la clé privée base64 dans un emplacement sûr car vous ne pourrez pas la récupérer ultérieurement.

```
{
    "keyPair": {
        "name": "MyPersonalKeyPair",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:KeyPair/55025c71-198f-403b-
b42f-a69433e724fb",
```

```
        "supportCode": "621291663362/MyPersonalKeyPair",
        "createdAt": 1569866556.567,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "KeyPair"
    },
    "publicKeyBase64": "ssh-rsa ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCV0xUEwx96amPERH7K1bVT1tTFl9OmNk6o7m5YVHk9xlOdMbDRbFvhtXvw4jz
+BHUgedGUXno6uF7agqxZNO1kPLJBIVTW26SSYBJ0tE
+y804UyVsjrbUqCaMXDhmfXpWulMPwuXhwcKh7e8hwoTfkiX0E6Ql
+KqF/MiA3w6DCjEqvvdIO7SiEZJFsuGNfYDDN3w60Rel5MUhmn3OJdn4y/
A7NWb3IxL4pPfVE4rgFRKU8n1jp9kwRnlVMVBOWuGXk6n+H6M2f1 ",
    "privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6\nb24xFDASBgNVBAsTC0lBTSBD
\nBgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
\nVQQHEwdTZWF0dGxlMQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
\nb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
\nYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
\nIbb3OhjZnzcvQAaREXAMPLEMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4\nnUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OEXAMPLELvjx79LjSTb
\nNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=\n-----END RSA PRIVATE KEY-----",
    "operation": {
        "id": "67f984db-9994-45fe-ad38-59bafcaf82ef",
        "resourceName": "MyPersonalKeyPair",
        "resourceType": "KeyPair",
        "createdAt": 1569866556.567,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationType": "CreateKeyPair",
        "status": "Succeeded",
        "statusChangedAt": 1569866556.704
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateKeyPairà la section Référence des AWS CLI commandes.

`create-load-balancer-tls-certificate`

L'exemple de code suivant montre comment utiliser`create-load-balancer-tls-certificate`.

AWS CLI

Pour créer un certificat TLS pour un équilibreur de charge

L'`create-load-balancer-tls-certificate`exemple suivant crée un certificat TLS attaché à l'équilibreur de charge spécifié. Le certificat créé s'applique aux domaines spécifiés. Remarque : Seuls deux certificats peuvent être créés pour un équilibreur de charge.

```
aws lightsail create-load-balancer-tls-certificate \
    --certificate-alternative-names abc.example.com \
    --certificate-domain-name example.com \
    --certificate-name MySecondCertificate \
    --load-balancer-name MyFirstLoadBalancer
```

Sortie :

```
{
    "operations": [
        {
            "id": "be663aed-cb46-41e2-9b23-e2f747245bd4",
            "resourceName": "MySecondCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569867364.971,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MyFirstLoadBalancer",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867365.219
        },
        {
            "id": "f3dfa930-969e-41cc-ac7d-337178716f6d",
```

```
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569867364.971,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MySecondCertificate",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867365.219
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateLoadBalancerTlsCertificateà la section Référence des AWS CLI commandes.

`create-load-balancer`

L'exemple de code suivant montre comment utiliser`create-load-balancer`.

AWS CLI

Pour créer un équilibreur de charge

L'`create-load-balancer`exemple suivant crée un équilibreur de charge avec un certificat TLS. Le certificat TLS s'applique aux domaines spécifiés et achemine le trafic vers les instances du port 80.

```
aws lightsail create-load-balancer \
    --certificate-alternative-names www.example.com test.example.com \
    --certificate-domain-name example.com \
    --certificate-name Certificate-1 \
    --instance-port 80 \
    --load-balancer-name LoadBalancer-1
```

Sortie :

```
{
    "operations": [
```

```
        {
            "id": "cc7b920a-83d8-4762-a74e-9174fe1540be",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1569867169.406,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1569867169.406
        },
        {
            "id": "658ed43b-f729-42f3-a8e4-3f8024d3c98d",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569867170.193,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "LoadBalancer-1",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867170.54
        },
        {
            "id": "4757a342-5181-4870-b1e0-227eebc35ab5",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1569867170.193,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "Certificate-1",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867170.54
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section relative aux équilibreurs de charge Lightsail dans le Guide du développeur de Lightsail.

- Pour plus de détails sur l'API, reportez-vous CreateLoadBalancerà la section Référence des AWS CLI commandes.

`create-relational-database-from-snapshot`

L'exemple de code suivant montre comment utiliser`create-relational-database-from-snapshot`.

AWS CLI

Pour créer une base de données gérée à partir d'un instantané

L'`create-relational-database-from-snapshot`exemple suivant crée une base de données gérée à partir de l'instantané spécifié dans la AWS région et la zone de disponibilité spécifiées, en utilisant le bundle de base de données standard de 15 USD. Remarque : Le bundle que vous spécifiez doit avoir des spécifications égales ou supérieures à celles du bundle de la base de données source d'origine utilisé pour créer le cliché.

```
aws lightsail create-relational-database-from-snapshot \
    --relational-database-snapshot-name Database-Oregon-1-1566839359 \
    --relational-database-name Database-1 \
    --availability-zone us-west-2a \
    --relational-database-bundle-id micro_1_0 \
    --no-publicly-accessible
```

Sortie :

```
{
    "operations": [
        {
            "id": "ad6d9193-9d5c-4ea1-97ae-8fe6de600b4c",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569867916.938,
            "location": {
```

```
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateRelationalDatabaseFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1569867918.643
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRelationalDatabaseFromSnapshotà la section Référence des AWS CLI commandes.

`create-relational-database-snapshot`

L'exemple de code suivant montre comment utiliser`create-relational-database-snapshot`.

AWS CLI

Pour créer un instantané d'une base de données gérée

L'`create-relational-database-snapshot`exemple suivant crée un instantané de la base de données gérée spécifiée.

```
aws lightsail create-relational-database-snapshot \
    --relational-database-name Database1 \
    --relational-database-snapshot-name RelationalDatabaseSnapshot1
```

Sortie :

```
{
    "operations": [
        {
            "id": "853667fb-ea91-4c02-8d20-8fc5fd43b9eb",
            "resourceName": "RelationalDatabaseSnapshot1",
            "resourceType": "RelationalDatabaseSnapshot",
            "createdAt": 1569868074.645,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "isTerminal": false,
            "operationDetails": "Database1",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569868074.645
        },
        {
            "id": "fbafa521-3cac-4be8-9773-1c143780b239",
            "resourceName": "Database1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569868074.645,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "RelationalDatabaseSnapshot1",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569868074.645
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRelationalDatabaseSnapshotà la section Référence des AWS CLI commandes.

`create-relational-database`

L'exemple de code suivant montre comment utiliser`create-relational-database`.

AWS CLI

Pour créer une base de données gérée

L'`create-relational-database`exemple suivant crée une base de données gérée dans la AWS région et la zone de disponibilité spécifiées, à l'aide du moteur de base de données MySQL 5.6 (mysql_5_6) et du bundle de base de données standard de 15 dollars américains (micro_1_0). La base de données gérée est préremplie d'un nom d'utilisateur principal et n'est pas accessible au public.

```
aws lightsail create-relational-database \
```

```
    --relational-database-name Database-1 \
    --availability-zone us-west-2a \
    --relational-database-blueprint-id mysql_5_6 \
    --relational-database-bundle-id micro_1_0 \
    --master-database-name dbmaster \
    --master-username user \
    --no-publicly-accessible
```

Sortie :

```
{
    "operations": [
        {
            "id": "b52bedee-73ed-4798-8d2a-9c12df89adcd",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569450017.244,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1569450018.637
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRelationalDatabaseà la section Référence des AWS CLI commandes.

`delete-auto-snapshot`

L'exemple de code suivant montre comment utiliser`delete-auto-snapshot`.

AWS CLI

Pour supprimer un instantané automatique

L'`delete-auto-snapshot`exemple suivant supprime l'instantané automatique `2019-10-10` de l'instance`WordPress-1`.

```
aws lightsail delete-auto-snapshot \
    --resource-name WordPress-1 \
    --date 2019-10-10
```

Sortie :

```
{
    "operations": [
        {
            "id": "31c36e09-3d52-46d5-b6d8-7EXAMPLE534a",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571088141.501,
            "location": {
                "availabilityZone": "us-west-2",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "DeleteAutoSnapshot-2019-10-10",
            "operationType": "DeleteAutoSnapshot",
            "status": "Succeeded"
        }
    ]
}
```

Pour plus d'informations, consultez la section Suppression des instantanés automatiques d'instances ou de disques dans Amazon Lightsail dans le guide de développement de Lightsail.

- Pour plus de détails sur l'API, reportez-vous DeleteAutoSnapshotà la section Référence des AWS CLI commandes.

`delete-disk-snapshot`

L'exemple de code suivant montre comment utiliser`delete-disk-snapshot`.

AWS CLI

Pour supprimer un instantané d'un disque de stockage par blocs

L'`delete-disk-snapshot`exemple suivant supprime l'instantané spécifié d'un disque de stockage en mode bloc.

```
aws lightsail delete-disk-snapshot \
    --disk-snapshot-name DiskSnapshot-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "d1e5766d-b81e-4595-ad5d-02afbccfcd5d",
            "resourceName": "DiskSnapshot-1",
            "resourceType": "DiskSnapshot",
            "createdAt": 1569873552.79,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteDiskSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569873552.79
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDiskSnapshotà la section Référence des AWS CLI commandes.

`delete-disk`

L'exemple de code suivant montre comment utiliser`delete-disk`.

AWS CLI

Pour supprimer un disque de stockage par blocs

L'`delete-disk`exemple suivant supprime le disque de stockage par blocs spécifié.

```
aws lightsail delete-disk \
    --disk-name Disk-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "6378c70f-4d75-4f7a-ab66-730fca0bb2fc",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569872887.864,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteDisk",
            "status": "Succeeded",
            "statusChangedAt": 1569872887.864
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDiskà la section Référence des AWS CLI commandes.

`delete-domain-entry`

L'exemple de code suivant montre comment utiliser`delete-domain-entry`.

AWS CLI

Pour supprimer une entrée de domaine (enregistrement DNS)

L'`delete-domain-entry`exemple suivant supprime l'entrée de domaine spécifiée d'un domaine existant.

Remarque : les opérations d'API liées au domaine de Lightsail ne sont disponibles que dans la région. `us-east-1` Si votre profil CLI est configuré pour utiliser une région différente, vous devez inclure le `--region us-east-1` paramètre, sinon la commande échoue.

```
aws lightsail delete-domain-entry \
    --region us-east-1 \
    --domain-name example.com \
    --domain-entry name=123.example.com,target=192.0.2.0,type=A
```

Sortie :

```
{
    "operation": {
        "id": "06eacd01-d785-420e-8daa-823150c7dca1",
        "resourceName": "example.com ",
        "resourceType": "Domain",
        "createdAt": 1569874157.005,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "DeleteDomainEntry",
        "status": "Succeeded",
        "statusChangedAt": 1569874157.005
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDomainEntryà la section Référence des AWS CLI commandes.

`delete-domain`

L'exemple de code suivant montre comment utiliser`delete-domain`.

AWS CLI

Pour supprimer un domaine (zone DNS)

L'`delete-domain`exemple suivant supprime le domaine spécifié et toutes les entrées du domaine (enregistrements DNS).

Remarque : les opérations d'API liées au domaine de Lightsail ne sont disponibles que dans la région. `us-east-1` Si votre profil CLI est configuré pour utiliser une région différente, vous devez inclure le `--region us-east-1` paramètre, sinon la commande échoue.

```
aws lightsail delete-domain \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "operation": {
        "id": "fcef5265-5af1-4a46-a3d7-90b5e18b9b32",
        "resourceName": "example.com",
        "resourceType": "Domain",
        "createdAt": 1569873788.13,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "DeleteDomain",
        "status": "Succeeded",
        "statusChangedAt": 1569873788.13
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDomainà la section Référence des AWS CLI commandes.

`delete-instance-snapshot`

L'exemple de code suivant montre comment utiliser`delete-instance-snapshot`.

AWS CLI

title

L'`delete-instance-snapshot`exemple suivant supprime le cliché spécifié d'une instance.

```
aws lightsail delete-instance-snapshot \
    --instance-snapshot-name WordPress-1-Snapshot-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "14dad182-976a-46c6-bfd4-9480482bf0ea",
            "resourceName": "WordPress-1-Snapshot-1",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1569874524.562,
```

```
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteInstanceSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569874524.562
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteInstanceSnapshotà la section Référence des AWS CLI commandes.

`delete-instance`

L'exemple de code suivant montre comment utiliser`delete-instance`.

AWS CLI

Pour supprimer une instance

L'`delete-instance`exemple suivant supprime l'instance spécifiée.

```
aws lightsail delete-instance \
    --instance-name WordPress-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "d77345a3-8f80-4d2e-b47d-aaa622718df2",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569874357.469,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
```

```
            "operationDetails": "WordPress-1",
            "operationType": "DetachDisk",
            "status": "Started",
            "statusChangedAt": 1569874357.469
        },
        {
            "id": "708fa606-2bfd-4e48-a2c1-0b856585b5b1",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569874357.465,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Disk-1",
            "operationType": "DetachDisk",
            "status": "Started",
            "statusChangedAt": 1569874357.465
        },
        {
            "id": "3187e823-8acb-405d-b098-fad5ceb17bec",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569874357.829,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteInstance",
            "status": "Succeeded",
            "statusChangedAt": 1569874357.829
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteInstanceà la section Référence des AWS CLI commandes.

`delete-key-pair`

L'exemple de code suivant montre comment utiliser`delete-key-pair`.

AWS CLI

Suppression d'une paire de clés

L'`delete-key-pair`exemple suivant supprime la paire de clés spécifiée.

```
aws lightsail delete-key-pair \
    --key-pair-name MyPersonalKeyPair
```

Sortie :

```
{
    "operation": {
        "id": "81621463-df38-4810-b866-6e801a15abbf",
        "resourceName": "MyPersonalKeyPair",
        "resourceType": "KeyPair",
        "createdAt": 1569874626.466,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationType": "DeleteKeyPair",
        "status": "Succeeded",
        "statusChangedAt": 1569874626.685
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteKeyPairà la section Référence des AWS CLI commandes.

`delete-known-host-keys`

L'exemple de code suivant montre comment utiliser`delete-known-host-keys`.

AWS CLI

Pour supprimer les clés d'hôte connues d'une instance

L'`delete-known-host-keys`exemple suivant supprime la clé d'hôte connue de l'instance spécifiée.

```
aws lightsail delete-known-host-keys \
    --instance-name Instance-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "c61afe9c-45a4-41e6-a97e-d212364da3f5",
            "resourceName": "Instance-1",
            "resourceType": "Instance",
            "createdAt": 1569874760.201,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteKnownHostKeys",
            "status": "Succeeded",
            "statusChangedAt": 1569874760.201
        }
    ]
}
```

Pour plus d'informations, consultez la section Résolution des problèmes de connexion avec le client SSH ou RDP basé sur le navigateur Amazon Lightsail dans le guide de développement de Lightsail.

- Pour plus de détails sur l'API, reportez-vous DeleteKnownHostKeysà la section Référence des AWS CLI commandes.

`delete-load-balancer-tls-certificate`

L'exemple de code suivant montre comment utiliser`delete-load-balancer-tls-certificate`.

AWS CLI

Pour supprimer un certificat TLS pour un équilibreur de charge

L'`delete-load-balancer-tls-certificate`exemple suivant supprime le certificat TLS spécifié de l'équilibreur de charge spécifié.

```
aws lightsail delete-load-balancer-tls-certificate \
    --load-balancer-name MyFirstLoadBalancer \
    --certificate-name MyFirstCertificate
```

Sortie :

```
{
    "operations": [
        {
            "id": "50bec274-e45e-4caa-8a69-b763ef636583",
            "resourceName": "MyFirstCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569874989.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569874989.48
        },
        {
            "id": "78c58cdc-a59a-4b27-8213-500638634a8f",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569874989.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569874989.48
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteLoadBalancerTlsCertificateà la section Référence des AWS CLI commandes.

`delete-load-balancer`

L'exemple de code suivant montre comment utiliser`delete-load-balancer`.

AWS CLI

Pour supprimer un équilibreur de charge

L'`delete-load-balancer`exemple suivant supprime l'équilibreur de charge spécifié et tous les certificats TLS associés.

```
aws lightsail delete-load-balancer \
    --load-balancer-name MyFirstLoadBalancer
```

Sortie :

```
{
    "operations": [
        {
            "id": "a8c968c7-72a3-4680-a714-af8f03eea535",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569875092.125,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1569875092.125
        },
        {
            "id": "f91a29fc-8ce3-4e69-a227-ea70ca890bf5",
            "resourceName": "MySecondCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569875091.938,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
```

```
            "status": "Started",
            "statusChangedAt": 1569875091.938
        },
        {
            "id": "cf64c060-154b-4eb4-ba57-84e2e41563d6",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569875091.94,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569875091.94
        }
    ]
}
```

Pour plus d'informations, voir le titre dans le guide.

- Pour plus de détails sur l'API, reportez-vous DeleteLoadBalancerà la section Référence des AWS CLI commandes.

`delete-relational-database-snapshot`

L'exemple de code suivant montre comment utiliser`delete-relational-database-snapshot`.

AWS CLI

Pour supprimer un instantané d'une base de données gérée

L'`delete-relational-database-snapshot`exemple suivant supprime l'instantané spécifié d'une base de données gérée.

```
aws lightsail delete-relational-database-snapshot \
    --relational-database-snapshot-name Database-Oregon-1-1566839359
```

Sortie :

```
{
    "operations": [
```

```
        {
            "id": "b99acae8-735b-4823-922f-30af580e3729",
            "resourceName": "Database-Oregon-1-1566839359",
            "resourceType": "RelationalDatabaseSnapshot",
            "createdAt": 1569875293.58,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteRelationalDatabaseSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569875293.58
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteRelationalDatabaseSnapshotà la section Référence des AWS CLI commandes.

`delete-relational-database`

L'exemple de code suivant montre comment utiliser`delete-relational-database`.

AWS CLI

Pour supprimer une base de données gérée

L'`delete-relational-database`exemple suivant supprime la base de données gérée spécifiée.

```
aws lightsail delete-relational-database \
    --relational-database-name Database-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "3b0c41c1-053d-46f0-92a3-14f76141dc86",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
```

```
            "createdAt": 1569875210.999,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1569875210.999
        },
        {
            "id": "01ddeae8-a87a-4a4b-a1f3-092c71bf9180",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569875211.029,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Database-1-FinalSnapshot-1569875210793",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569875211.029
        },
        {
            "id": "74d73681-30e8-4532-974e-1f23cd3f9f73",
            "resourceName": "Database-1-FinalSnapshot-1569875210793",
            "resourceType": "RelationalDatabaseSnapshot",
            "createdAt": 1569875211.029,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Database-1",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569875211.029
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteRelationalDatabaseà la section Référence des AWS CLI commandes.

`detach-static-ip`

L'exemple de code suivant montre comment utiliser`detach-static-ip`.

AWS CLI

Pour détacher une adresse IP statique d'une instance

L'`detach-static-ip`exemple suivant détache l'adresse IP statique `StaticIp-1` de toute instance attachée.

```
aws lightsail detach-static-ip \
    --static-ip-name StaticIp-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "2a43d8a3-9f2d-4fe7-bdd0-eEXAMPLE3cf3",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571088261.999,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "DetachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571088261.999
        },
        {
            "id": "41a7d40c-74e8-4d2e-a837-cEXAMPLEf747",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571088262.022,
            "location": {
```

```
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "StaticIp-1",
            "operationType": "DetachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571088262.022
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DetachStaticIpà la section Référence des AWS CLI commandes.

`get-active-names`

L'exemple de code suivant montre comment utiliser`get-active-names`.

AWS CLI

Pour obtenir les noms des ressources actives

L'`get-active-names`exemple suivant renvoie les noms des ressources actives dans la AWS région configurée.

```
aws lightsail get-active-names
```

Sortie :

```
{
    "activeNames": [
        "WordPress-1",
        "StaticIp-1",
        "MEAN-1",
        "Plesk_Hosting_Stack_on_Ubuntu-1"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetActiveNamesà la section Référence des AWS CLI commandes.

`get-auto-snapshots`

L'exemple de code suivant montre comment utiliser`get-auto-snapshots`.

AWS CLI

Pour obtenir les instantanés automatiques disponibles pour une instance

L'`get-auto-snapshots`exemple suivant renvoie par exemple les instantanés automatiques disponibles. `WordPress-1`

```
aws lightsail get-auto-snapshots \
    --resource-name WordPress-1
```

Sortie :

```
{
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "autoSnapshots": [
        {
            "date": "2019-10-14",
            "createdAt": 1571033872.0,
            "status": "Success",
            "fromAttachedDisks": []
        },
        {
            "date": "2019-10-13",
            "createdAt": 1570947473.0,
            "status": "Success",
            "fromAttachedDisks": []
        },
        {
            "date": "2019-10-12",
            "createdAt": 1570861072.0,
            "status": "Success",
            "fromAttachedDisks": []
        },
        {
            "date": "2019-10-11",
            "createdAt": 1570774672.0,
            "status": "Success",
            "fromAttachedDisks": []
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Conservation des instantanés automatiques des instances ou des disques dans Amazon Lightsail dans le Guide de développement de Lightsail.

- Pour plus de détails sur l'API, reportez-vous GetAutoSnapshotsà la section Référence des AWS CLI commandes.

`get-blueprints`

L'exemple de code suivant montre comment utiliser`get-blueprints`.

AWS CLI

Pour obtenir les plans des nouvelles instances

L'`get-blueprints`exemple suivant affiche les détails de tous les plans disponibles qui peuvent être utilisés pour créer de nouvelles instances dans Amazon Lightsail.

```
aws lightsail get-blueprints
```

Sortie :

```
{
    "blueprints": [
        {
            "blueprintId": "wordpress",
            "name": "WordPress",
            "group": "wordpress",
            "type": "app",
            "description": "Bitnami, the leaders in application packaging, and
 Automattic, the experts behind WordPress, have teamed up to offer this official
 WordPress image. This image is a pre-configured, ready-to-run image for running
 WordPress on Amazon Lightsail. WordPress is the world's most popular content
 management platform. Whether it's for an enterprise or small business website, or
 a personal or corporate blog, content authors can easily create content using its
 new Gutenberg editor, and developers can extend the base platform with additional
 features. Popular plugins like Jetpack, Akismet, All in One SEO Pack, WP Mail,
 Google Analytics for WordPress, and Amazon Polly are all pre-installed in this
 image. Let's Encrypt SSL certificates are supported through an auto-configuration
 script.",
```

```
            "isActive": true,
            "minPower": 0,
            "version": "6.5.3-0",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U#pdp-
usage",
            "platform": "LINUX_UNIX"
        },
        {
            "blueprintId": "lamp_8_bitnami",
            "name": "LAMP (PHP 8)",
            "group": "lamp_8",
            "type": "app",
            "description": "LAMP with PHP 8.X packaged by Bitnami enables you
 to quickly start building your websites and applications by providing a coding
 framework. As a developer, it provides standalone project directories to store your
 applications. This blueprint is configured for production environments. It includes
 SSL auto-configuration with Let's Encrypt certificates, and the latest releases of
 PHP, Apache, and MariaDB on Linux. This application also includes phpMyAdmin, PHP
 main modules and Composer.",
            "isActive": true,
            "minPower": 0,
            "version": "8.2.18-4",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu#pdp-usage",
            "platform": "LINUX_UNIX"
        },
        {
            "blueprintId": "nodejs",
            "name": "Node.js",
            "group": "node",
            "type": "app",
            "description": "Node.js packaged by Bitnami is a pre-configured, ready
 to run image for Node.js on Amazon EC2. It includes the latest version of Node.js,
 Apache, Python and Redis. The image supports multiple Node.js applications, each
 with its own virtual host and project directory. It is configured for production
 use and is secure by default, as all ports except HTTP, HTTPS and SSH ports are
 closed. Let's Encrypt SSL certificates are supported through an auto-configuration
 script. Developers benefit from instant access to a secure, update and consistent
```

```
 Node.js environment without having to manually install and configure multiple
 components and libraries.",
            "isActive": true,
            "minPower": 0,
            "version": "18.20.2-0",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAKO",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAKO#pdp-
usage",
            "platform": "LINUX_UNIX"
        },
        ...
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBlueprintsà la section Référence des AWS CLI commandes.

`get-bundles`

L'exemple de code suivant montre comment utiliser`get-bundles`.

AWS CLI

Pour obtenir les offres groupées pour les nouvelles instances

L'`get-bundles`exemple suivant affiche les détails de tous les ensembles disponibles qui peuvent être utilisés pour créer de nouvelles instances dans Amazon Lightsail.

```
aws lightsail get-bundles
```

Sortie :

```
{
    "bundles": [
        {
            "price": 5.0,
            "cpuCount": 2,
            "diskSizeInGb": 20,
            "bundleId": "nano_3_0",
            "instanceType": "nano",
```

```
            "isActive": true,
            "name": "Nano",
            "power": 298,
            "ramSizeInGb": 0.5,
            "transferPerMonthInGb": 1024,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        {
            "price": 7.0,
            "cpuCount": 2,
            "diskSizeInGb": 40,
            "bundleId": "micro_3_0",
            "instanceType": "micro",
            "isActive": true,
            "name": "Micro",
            "power": 500,
            "ramSizeInGb": 1.0,
            "transferPerMonthInGb": 2048,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        {
            "price": 12.0,
            "cpuCount": 2,
            "diskSizeInGb": 60,
            "bundleId": "small_3_0",
            "instanceType": "small",
            "isActive": true,
            "name": "Small",
            "power": 1000,
            "ramSizeInGb": 2.0,
            "transferPerMonthInGb": 3072,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        ...
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBundlesà la section Référence des AWS CLI commandes.

`get-cloud-formation-stack-records`

L'exemple de code suivant montre comment utiliser`get-cloud-formation-stack-records`.

AWS CLI

Pour obtenir les enregistrements des CloudFormation piles et leurs piles associées

L'`get-cloud-formation-stack-records`exemple suivant affiche des détails sur les enregistrements de CloudFormation pile et leurs piles associées utilisés pour créer des EC2 ressources Amazon à partir de snapshots Amazon Lightsail exportés.

```
aws lightsail get-cloud-formation-stack-records
```

Sortie :

```
{
    "cloudFormationStackRecords": [
        {
            "name": "CloudFormationStackRecord-588a4243-
e2d1-490d-8200-3a7513ecebdf",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:CloudFormationStackRecord/28d646ab-27bc-48d9-a422-1EXAMPLE6d37",
            "createdAt": 1565301666.586,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "CloudFormationStackRecord",
            "state": "Succeeded",
            "sourceInfo": [
                {
                    "resourceType": "ExportSnapshotRecord",
                    "name": "ExportSnapshotRecord-
e02f23d7-0453-4aa9-9c95-91aa01a141dd",
                    "arn": "arn:aws:lightsail:us-
west-2:111122223333:ExportSnapshotRecord/f12b8792-f3ea-4d6f-b547-2EXAMPLE8796"
                }
            ],
```

```
            "destinationInfo": {
                "id": "arn:aws:cloudformation:us-west-2:111122223333:stack/
Lightsail-Stack-588a4243-e2d1-490d-8200-3EXAMPLEebdf/063203b0-
ba28-11e9-838b-0EXAMPLE8b00",
                "service": "Aws::CloudFormation::Stack"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetCloudFormationStackRecordsà la section Référence des AWS CLI commandes.

`get-disk-snapshot`

L'exemple de code suivant montre comment utiliser`get-disk-snapshot`.

AWS CLI

Pour obtenir des informations sur un instantané de disque

L'`get-disk-snapshot`exemple suivant affiche des détails sur le snapshot du disque`Disk-1-1566839161`.

```
aws lightsail get-disk-snapshot \
    --disk-snapshot-name Disk-1-1566839161
```

Sortie :

```
{
    "diskSnapshot": {
        "name": "Disk-1-1566839161",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",
        "supportCode": "6EXAMPLE3362/snap-0EXAMPLE06100d09",
        "createdAt": 1566839163.749,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "DiskSnapshot",
        "tags": [],
```

```
        "sizeInGb": 8,
        "state": "completed",
        "progress": "100%",
        "fromDiskName": "Disk-1",
        "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
        "isFromAutoSnapshot": false
    }
}
```

Pour plus d'informations, voir le titre dans le guide.

- Pour plus de détails sur l'API, reportez-vous GetDiskSnapshotà la section Référence des AWS CLI commandes.

`get-disk-snapshots`

L'exemple de code suivant montre comment utiliser`get-disk-snapshots`.

AWS CLI

Pour obtenir des informations sur tous les instantanés de disque

L'`get-disk-snapshots`exemple suivant affiche les détails de tous les instantanés de disque dans la AWS région configurée.

```
aws lightsail get-disk-snapshots
```

Sortie :

```
{
    "diskSnapshots": [
        {
            "name": "Disk-2-1571090588",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:DiskSnapshot/32e889a9-38d4-4687-9f21-eEXAMPLE7839",
            "supportCode": "6EXAMPLE3362/snap-0EXAMPLE1ca192a4",
            "createdAt": 1571090591.226,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "resourceType": "DiskSnapshot",
            "tags": [],
            "sizeInGb": 8,
            "state": "completed",
            "progress": "100%",
            "fromDiskName": "Disk-2",
            "fromDiskArn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
            "isFromAutoSnapshot": false
        },
        {
            "name": "Disk-1-1566839161",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",
            "supportCode": "6EXAMPLE3362/snap-0EXAMPLEe06100d09",
            "createdAt": 1566839163.749,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "DiskSnapshot",
            "tags": [],
            "sizeInGb": 8,
            "state": "completed",
            "progress": "100%",
            "fromDiskName": "Disk-1",
            "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
            "isFromAutoSnapshot": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDiskSnapshotsà la section Référence des AWS CLI commandes.

`get-disk`

L'exemple de code suivant montre comment utiliser`get-disk`.

AWS CLI

Pour obtenir des informations sur un disque de stockage par blocs

L'`get-disk`exemple suivant affiche des informations détaillées sur le disque`Disk-1`.

```
aws lightsail get-disk \
    --disk-name Disk-1
```

Sortie :

```
{
    "disk": {
        "name": "Disk-1",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
        "createdAt": 1566585439.587,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Disk",
        "tags": [],
        "sizeInGb": 8,
        "isSystemDisk": false,
        "iops": 100,
        "path": "/dev/xvdf",
        "state": "in-use",
        "attachedTo": "WordPress_Multisite-1",
        "isAttached": true,
        "attachmentState": "attached"
    }
}
```

Pour plus d'informations, voir le titre dans le guide.

- Pour plus de détails sur l'API, reportez-vous GetDiskà la section Référence des AWS CLI commandes.

`get-disks`

L'exemple de code suivant montre comment utiliser`get-disks`.

AWS CLI

Pour obtenir des informations sur tous les disques de stockage par blocs

L'`get-disks`exemple suivant affiche les détails de tous les disques de la AWS région configurée.

```
aws lightsail get-disks
```

Sortie :

```
{
    "disks": [
        {
            "name": "Disk-2",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
            "supportCode": "6EXAMPLE3362/vol-0EXAMPLE929602087",
            "createdAt": 1571090461.634,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Disk",
            "tags": [],
            "sizeInGb": 8,
            "isSystemDisk": false,
            "iops": 100,
            "state": "available",
            "isAttached": false,
            "attachmentState": "detached"
        },
        {
            "name": "Disk-1",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
            "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
            "createdAt": 1566585439.587,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Disk",
            "tags": [],
            "sizeInGb": 8,
            "isSystemDisk": false,
            "iops": 100,
            "path": "/dev/xvdf",
```

```
            "state": "in-use",
            "attachedTo": "WordPress_Multisite-1",
            "isAttached": true,
            "attachmentState": "attached"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDisksà la section Référence des AWS CLI commandes.

`get-domain`

L'exemple de code suivant montre comment utiliser`get-domain`.

AWS CLI

Pour obtenir des informations sur un domaine

L'`get-domain`exemple suivant affiche les détails du domaine`example.com`.

Remarque : les opérations d'API liées au domaine de Lightsail ne sont disponibles que dans la région. `us-east-1` AWS Si votre profil CLI est configuré pour utiliser une région différente, vous devez inclure le paramètre`` --region us-east-1``, sinon la commande échouera.

```
aws lightsail get-domain \
    --domain-name example.com \
    --region us-east-1
```

Sortie :

```
{
    "domain": {
        "name": "example.com",
        "arn":
 "arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",
        "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",
        "createdAt": 1570728588.6,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
```

```
        "resourceType": "Domain",
        "tags": [],
        "domainEntries": [
            {
                "id": "-1682899164",
                "name": "example.com",
                "target": "192.0.2.0",
                "isAlias": false,
                "type": "A"
            },
            {
                "id": "1703104243",
                "name": "example.com",
                "target": "ns-137.awsdns-17.com",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "-1038331153",
                "name": "example.com",
                "target": "ns-1710.awsdns-21.co.uk",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "-2107289565",
                "name": "example.com",
                "target": "ns-692.awsdns-22.net",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "1582095705",
                "name": "example.com",
                "target": "ns-1436.awsdns-51.org",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "-1769796132",
                "name": "example.com",
                "target": "ns-1710.awsdns-21.co.uk. awsdns-hostmaster.amazon.com. 1
7200 900 1209600 86400",
                "isAlias": false,
```

```
                "type": "SOA"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDomainà la section Référence des AWS CLI commandes.

`get-domains`

L'exemple de code suivant montre comment utiliser`get-domains`.

AWS CLI

Pour obtenir des informations sur tous les domaines

L'`get-domains`exemple suivant affiche les détails de tous les domaines de la AWS région configurée.

Remarque : les opérations d'API liées au domaine de Lightsail ne sont disponibles que dans la région. `us-east-1` AWS Si votre profil CLI est configuré pour utiliser une région différente, vous devez inclure le `--region us-east-1` paramètre, sinon la commande échoue.

```
aws lightsail get-domains \
    --region us-east-1
```

Sortie :

```
{
    "domains": [
        {
            "name": "example.com",
            "arn":
 "arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",
            "createdAt": 1570728588.6,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
```

```
            "tags": [],
            "domainEntries": [
                {
                    "id": "-1682899164",
                    "name": "example.com",
                    "target": "192.0.2.0",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "1703104243",
                    "name": "example.com",
                    "target": "ns-137.awsdns-17.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-1038331153",
                    "name": "example.com",
                    "target": "ns-4567.awsdns-21.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-2107289565",
                    "name": "example.com",
                    "target": "ns-333.awsdns-22.net",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1582095705",
                    "name": "example.com",
                    "target": "ns-1111.awsdns-51.org",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-1769796132",
                    "name": "example.com",
                    "target": "ns-1234.awsdns-21.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
```

```
                },
                {
                    "id": "1029454894",
                    "name": "_dead6a124ede046a0319eb44a4eb3cbc.example.com",
                    "target": "_be133b0a0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws",
                    "isAlias": false,
                    "type": "CNAME"
                }
            ]
        },
        {
            "name": "example.net",
            "arn": "arn:aws:lightsail:global:111122223333:Domain/9c9f0d70-
c92e-4753-86c2-6EXAMPLE029d",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLE5TPKMV",
            "createdAt": 1556661071.384,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
            "domainEntries": [
                {
                    "id": "-766320943",
                    "name": "example.net",
                    "target": "192.0.2.2",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "-453913825",
                    "name": "example.net",
                    "target": "ns-123.awsdns-10.net",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1553601564",
                    "name": "example.net",
                    "target": "ns-4444.awsdns-47.co.uk",
                    "isAlias": false,
                    "type": "NS"
```

```
                },
                {
                    "id": "1653797661",
                    "name": "example.net",
                    "target": "ns-7890.awsdns-61.org",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "706414698",
                    "name": "example.net",
                    "target": "ns-123.awsdns-44.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "337271745",
                    "name": "example.net",
                    "target": "ns-4444.awsdns-47.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
                {
                    "id": "-1785431096",
                    "name": "www.example.net",
                    "target": "192.0.2.2",
                    "isAlias": false,
                    "type": "A"
                }
            ]
        },
        {
            "name": "example.org",
            "arn": "arn:aws:lightsail:global:111122223333:Domain/
f0f13ba3-3df0-4fdc-8ebb-1EXAMPLEf26e",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEAFO38",
            "createdAt": 1556661199.106,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
```

```
            "domainEntries": [
                {
                    "id": "2065301345",
                    "name": "example.org",
                    "target": "192.0.2.4",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "-447198516",
                    "name": "example.org",
                    "target": "ns-123.awsdns-45.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "136463022",
                    "name": "example.org",
                    "target": "ns-9999.awsdns-15.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1395941679",
                    "name": "example.org",
                    "target": "ns-555.awsdns-01.net",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "872052569",
                    "name": "example.org",
                    "target": "ns-6543.awsdns-38.org",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1001949377",
                    "name": "example.org",
                    "target": "ns-1234.awsdns-15.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
```

```
                {
                    "id": "1046191192",
                    "name": "www.example.org",
                    "target": "192.0.2.4",
                    "isAlias": false,
                    "type": "A"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDomainsà la section Référence des AWS CLI commandes.

`get-export-snapshot-record`

L'exemple de code suivant montre comment utiliser`get-export-snapshot-record`.

AWS CLI

Pour obtenir les enregistrements des instantanés exportés vers Amazon EC2

L'`get-export-snapshot-record`exemple suivant affiche des informations sur les instantanés d'instance ou de disque Amazon Lightsail exportés vers Amazon. EC2

```
aws lightsail get-export-snapshot-records
```

Sortie :

```
{
    "exportSnapshotRecords": [
        {
            "name": "ExportSnapshotRecord-d2da10ce-0b3c-4ae1-ab3a-2EXAMPLEa586",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:ExportSnapshotRecord/076c7060-b0cc-4162-98f0-2EXAMPLEe28e",
            "createdAt": 1543534665.678,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "ExportSnapshotRecord",
```

```
            "state": "Succeeded",
            "sourceInfo": {
                "resourceType": "InstanceSnapshot",
                "createdAt": 1540339310.706,
                "name": "WordPress-512MB-Oregon-1-1540339219",
                "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/5446f534-ed60-4c17-b4a5-bEXAMPLEf8b7",
                "fromResourceName": "WordPress-512MB-Oregon-1",
                "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/4b8f1f24-e4d1-4cf3-88ff-cEXAMPLEa397",
                "instanceSnapshotInfo": {
                    "fromBundleId": "nano_2_0",
                    "fromBlueprintId": "wordpress_4_9_8",
                    "fromDiskInfo": [
                        {
                            "path": "/dev/sda1",
                            "sizeInGb": 20,
                            "isSystemDisk": true
                        }
                    ]
                }
            },
            "destinationInfo": {
                "id": "ami-0EXAMPLEc0d65058e",
                "service": "Aws::EC2::Image"
            }
        },
        {
            "name": "ExportSnapshotRecord-1c94e884-40ff-4fe1-9302-0EXAMPLE14c2",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:ExportSnapshotRecord/
fb392ce8-6567-4013-9bfd-3EXAMPLE5b4c",
            "createdAt": 1543432110.2,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "ExportSnapshotRecord",
            "state": "Succeeded",
            "sourceInfo": {
                "resourceType": "InstanceSnapshot",
                "createdAt": 1540833603.545,
                "name": "LAMP_PHP_5-512MB-Oregon-1-1540833565",
                "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/82334399-b5f2-49ec-8382-0EXAMPLEe45f",
```

```
                "fromResourceName": "LAMP_PHP_5-512MB-Oregon-1",
                "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/863b9f35-ab1e-4418-bdd2-1EXAMPLEbab2",
                "instanceSnapshotInfo": {
                    "fromBundleId": "nano_2_0",
                    "fromBlueprintId": "lamp_5_6_37_2",
                    "fromDiskInfo": [
                        {
                            "path": "/dev/sda1",
                            "sizeInGb": 20,
                            "isSystemDisk": true
                        }
                    ]
                }
            },
            "destinationInfo": {
                "id": "ami-0EXAMPLE7c5ec84e2",
                "service": "Aws::EC2::Image"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetExportSnapshotRecordà la section Référence des AWS CLI commandes.

`get-instance-access-details`

L'exemple de code suivant montre comment utiliser`get-instance-access-details`.

AWS CLI

Pour obtenir des informations sur la clé d'hôte pour une instance

L'`get-instance-access-details`exemple suivant affiche les informations clés de l'hôte, par exemple`WordPress_Multisite-1`.

```
aws lightsail get-instance-access-details \
    --instance-name WordPress_Multisite-1
```

Sortie :

```
{
    "accessDetails": {
        "certKey": "ssh-rsa-cert-v01@openssh.com
 AEXAMPLEaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
vEXAMPLE2eBWJyQvn7ol/
i0+s966h5sx8qUD79lPB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjblLj26Fb37EKda9HVfCOu8pWbvky7Tyn9w29
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/
K1EXAMPLEAAAAAAAABAAAALS00MzMzMDU4MzA4ODg1MTY2NjM4Onp6UWlndHk4UElRSG9STitOTG5QSEE9PQAAAAsAAA
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN
+tiXm7Y8OgziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLuV7cnWPB7iUlHQg
+E3LUKrV4ZFw9pj7X2dFdNKfMxwWgI1ISWKimEXAMPLEeHjrf1Rqc/
QH6TpWCvPfcx8uvwVqdwTfkE/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/
KwmM7pCwJbSLDMO7n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEobqfdtyYJHHel1EyyEJs1fWNU3D5JIGlgzcPAV
+ZlbQyUCZXf0oslSa+HE85fO/
FRq9SVSBSHrmbeb0frlPhgMzgSmqLeyhlbr6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPWvRgcr0qI2EL0tAajnpQQ8UZq
Aqter0xN5PhFL0J49OWTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8clll327EXAMPLE/
e89GC89KcmKCxfGQniDAUgF8UqofIbq3ZOUgiAAYCVXclI4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/
Oz0cTR+4sAAAIPAAAAB3NzaC1yc2EAAAIAQnG/
L0DqiSnLrWhEox4aHqMgd0m0oLLAYx6OQH9F0TM9EXAMPLE961rzSCMon7ZgsWNnL0OwZQgDG
+rtJ4N0B7HOVwns4ynUFbzNQ3qFGGeE3lKwX1L41vV1iSy7sDk8aI0LmrKJi1LE1Qc1l8uboRlwoXOYEXAMPLEaUCeX
+10+WEXAMPLEg6Y4U4ZvE2B3xyRdpvysb5TGFNtk5qPslacnVkoLOGsZZXMpLGJnG4OBpQLLtpj9sNMxAgZPCAUjhkqk
+nxO9O4NUZ2pTWbVSUaV1gm6pug9xbwNO1Im21t34JeLlKTqxcJ6zzS8W0c0KKpAm5c4hWkseMbyutS2jav/4hiS
+BhrYgptzfwe5qRXEXAMPLEHZQr3YfGzYoBJ/
lLK3NHhxOihhsfAYwMei0BFZT1F/7CT3IH4iitEkIgodi06/
Mw6UDqMPozyQCK1lEA6LFhYCOZG9drWcoRa74lM4kY9TP028Za8gDMh1WpkXLq9Gixon5OHP8aM/
sEXAMPLEr2+fnkw+1BtoO5L6+VKoPlXaGqZ/fBYEXAMPLEAMQHjnLM1JYNvtEEPhp+TNzXHzuixWf/
Ht04m0AVpXrzIDXaS1O2tXY=",
        "ipAddress": "192.0.2.0",
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nEXAMPLEBAAKCAQEA+AD3qeU2toBy5O5v7wnRLVo/tngVickL5+6Jf4tPrPeuoebM
\nfKlA+/ZTwe6uVBEneVWRhbcra8pH0CZ44sKnuxFeWoM7425S49uhW9+xCnWvR1Xw
\njrvKVm75Mu08p/cNvfWugrBuaPB65DspgxNnOfZWMVxpIpSqOSPWmSwQHV597d6C
\nrEXAMPLEo8hJmqz2KFQ09X7fB2lBruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaFl0GNGQ8qy1HGUiscquZ9NDIL49n4mXbfsTH
\n0EXAMPLE12ZqsfLiYnSaUYCwjE74qH8ECVPytQIDAQABAoIBAHeZV9Z58JHAjifz
\nCEXAMPLEEqC3doOVDgXSlkKI92qNo4z2VcUEho878paCuVVXVHcCGgSnGeyIh2tN
\nMEXAMPLESohR427BhH3YLA+3Z5SIvnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn
\nIEXAMPLEM93oF9eSZCjcLKB/jGHsfb0eCDMP8BshHE2beuqzVMoK1DxOnvoP3+Fp
\nAEXAMPLESq6pDpCo9YVUX8g1u3Ro9cPl2LXHDy+oVEY5KhbZQJ7VU1I72WOvppWW
\nOEXAMPLEkgYlq7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoNONQ9ZPaWHyJvymeud
\noQTNuz0CgYEA/LFWNTEZrzdzdR1kJmyNRmAermU0B6utyNENChAlHGSHkB+1lVSh
\nbEXAMPLEQo9ooUeW5UxO3YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob
\n8l7CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/EOrB0we0RKyDSzWScCgYEA+z/r
```

```
\niob+nJZq0YbnOSuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJdONKDdkuiOd52aAL
\nVgn7nLo88rVWKhJwVc4tu/rNgZLcR3bP4+kL6zand0KQnMLyOzNA2Ys26aa5udH1\nqWl0WTt9WEm/
h10ndC1knOMectrvsG17b38y5sMCgYEA54NiRGGz8oCPW6GN/FZA
\nKEXAMPLE5tw34GEH3Uxlc9n3CejDaQmczOATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkc8Z71L/
agZEAaVCEog9FqfSqwB
+XTfoKh8qur74X1yCu9p6gof1q6k9\neEXAMPLEchJcNNOg4ETIfMkCgYBdVORRhE4mqvWpOdzA7v66FdEz2YSkjAXKk
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\ngEXAMPLEBogjw4MqHKL1bPKMHyQU8/
q24PaYgzHPzy13wlH6pTYf1XqlHdE2D6Vv\nyEXAMPLEgQC3i/
kVVhky/2XRwRVlC7JO2Bg3QGTx38hpmDa5IuofKANjA+Wa3/zy\nbEXAMPLE6ytQgD9GN/YtBq+uhO
+2ZkvXPL+CWRi0ZRXpPwYDBBFU9Cw0AuWWGlL8\nwEXAMPLExMlcysRgcWB9RNgf3AuOpFd2i6XT/
riNsvvkpmJ+VooU8g==\n-----END RSA PRIVATE KEY-----\n",
        "protocol": "ssh",
        "instanceName": "WordPress_Multisite-1",
        "username": "bitnami",
        "hostKeys": [
            {
                "algorithm": "ssh-rsa",
                "publicKey":
 "AEXAMPLEaC1yc2EAAAADAQABAAABAQCoeR9ieZTjQ3pXCHczuAYZFjlF7t
+uBkXuqeGMRex78pCvmS+DiEXAMPLEuJ1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdHFzy3guP
+BKclWqtxJEXAMPLEsBGqZZlrIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0XlMs7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHPp44Og36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
Ot6y9HwvykEXAMPLEAfbKjbR42+u6+OSlkr4d339q2U1sTDytJhhs8HUel1wTfGRfp",
                "witnessedAt": 1570744377.699,
                "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
                "fingerprintSHA256": "SHA256:IEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
            },
            {
                "algorithm": "ssh-ed25519",
                "publicKey":
 "AEXAMPLEaC1lZDI1NTE5AAAAIC1gwGPDfGaONxEXAMPLEJX3UNap781QxHQmn8nzlrUv",
                "witnessedAt": 1570744377.697,
                "fingerprintSHA1": "SHA1:VEXAMPLE5ReqSmTgv03sSUw9toU",
                "fingerprintSHA256": "SHA256:0EXAMPLEdE6tI95k3TJpG
+qhJbAoknB0yz9nAEaDt3A"
            },
            {
                "algorithm": "ecdsa-sha2-nistp256",
                "publicKey":
 "AEXAMPLEZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABEXAMPLE9B4mZy8YSsZW7cixCDq5yHSAAxjJkDo5
+EnKlDCsYtUkxxEXAMPLE6VOWL2z63RTKa2AUPgd8irjxWI=",
                "witnessedAt": 1570744377.707,
```

```
                "fingerprintSHA1": "SHA1:UEXAMPLEOYCfXsCf2G6tDg+7YG0",
                "fingerprintSHA256": "SHA256:wEXAMPLEQ9a/
iEXAMPLEhRufm6U9vFU4cpkMPHnBsNA"
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstanceAccessDetailsà la section Référence des AWS CLI commandes.

`get-instance-metric-data`

L'exemple de code suivant montre comment utiliser`get-instance-metric-data`.

AWS CLI

Pour obtenir des données métriques pour une instance

L'`get-instance-metric-data`exemple suivant renvoie le pourcentage moyen de `CPUUtilization` toutes les `7200` secondes (2 heures) entre `1571342400` et par `1571428800` exemple`MEAN-1`.

Nous vous recommandons d'utiliser un convertisseur de temps Unix pour identifier les heures de début et de fin.

```
aws lightsail get-instance-metric-data \
    --instance-name MEAN-1 \
    --metric-name CPUUtilization \
    --period 7200 \
    --start-time 1571342400 \
    --end-time 1571428800 \
    --unit Percent \
    --statistics Average
```

Sortie :

```
{
    "metricName": "CPUUtilization",
    "metricData": [
        {
            "average": 0.26113718770120725,
```

```
            "timestamp": 1571342400.0,
            "unit": "Percent"
        },
        {
            "average": 0.26861268928111953,
            "timestamp": 1571392800.0,
            "unit": "Percent"
        },
        {
            "average": 0.28187475104748777,
            "timestamp": 1571378400.0,
            "unit": "Percent"
        },
        {
            "average": 0.2651936960458352,
            "timestamp": 1571421600.0,
            "unit": "Percent"
        },
        {
            "average": 0.2561856213712188,
            "timestamp": 1571371200.0,
            "unit": "Percent"
        },
        {
            "average": 0.3021383254607764,
            "timestamp": 1571356800.0,
            "unit": "Percent"
        },
        {
            "average": 0.2618381649223539,
            "timestamp": 1571407200.0,
            "unit": "Percent"
        },
        {
            "average": 0.26331929394825787,
            "timestamp": 1571400000.0,
            "unit": "Percent"
        },
        {
            "average": 0.2576348407007818,
            "timestamp": 1571385600.0,
            "unit": "Percent"
        },
        {
```

```
            "average": 0.2513008454658378,
            "timestamp": 1571364000.0,
            "unit": "Percent"
        },
        {
            "average": 0.26329974562758346,
            "timestamp": 1571414400.0,
            "unit": "Percent"
        },
        {
            "average": 0.2667092536656445,
            "timestamp": 1571349600.0,
            "unit": "Percent"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstanceMetricDataà la section Référence des AWS CLI commandes.

`get-instance-port-states`

L'exemple de code suivant montre comment utiliser`get-instance-port-states`.

AWS CLI

Pour obtenir des informations sur le pare-feu d'une instance

L'`get-instance-port-states`exemple suivant renvoie les ports de pare-feu configurés par exemple`MEAN-1`.

```
aws lightsail get-instance-port-states \
    --instance-name MEAN-1
```

Sortie :

```
{
    "portStates": [
        {
            "fromPort": 80,
            "toPort": 80,
            "protocol": "tcp",
```

```
            "state": "open"
        },
        {
            "fromPort": 22,
            "toPort": 22,
            "protocol": "tcp",
            "state": "open"
        },
        {
            "fromPort": 443,
            "toPort": 443,
            "protocol": "tcp",
            "state": "open"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstancePortStatesà la section Référence des AWS CLI commandes.

`get-instance-snapshot`

L'exemple de code suivant montre comment utiliser`get-instance-snapshot`.

AWS CLI

Pour obtenir des informations sur un instantané d'instance spécifié

L'`get-instance-snapshot`exemple suivant affiche des détails sur le cliché d'instance spécifié.

```
aws lightsail get-instance-snapshot \
    --instance-snapshot-name MEAN-1-1571419854
```

Sortie :

```
{
    "instanceSnapshot": {
        "name": "MEAN-1-1571419854",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
        "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
        "createdAt": 1571419891.927,
```

```
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "InstanceSnapshot",
        "tags": [],
        "state": "available",
        "fromAttachedDisks": [],
        "fromInstanceName": "MEAN-1",
        "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
        "fromBlueprintId": "mean",
        "fromBundleId": "medium_3_0",
        "isFromAutoSnapshot": false,
        "sizeInGb": 80
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstanceSnapshotà la section Référence des AWS CLI commandes.

`get-instance-snapshots`

L'exemple de code suivant montre comment utiliser`get-instance-snapshots`.

AWS CLI

Pour obtenir des informations sur tous les instantanés de votre instance

L'`get-instance-snapshots`exemple suivant affiche des détails sur tous les instantanés d'instance dans la AWS région configurée.

```
aws lightsail get-instance-snapshots
```

Sortie :

```
{
    "instanceSnapshots": [
        {
            "name": "MEAN-1-1571421498",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
a20e6ebe-b0ee-4ae4-a750-3EXAMPLEcb0c",
```

```
            "supportCode": "6EXAMPLE3362/ami-0EXAMPLEe33cabfa1",
            "createdAt": 1571421527.755,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "InstanceSnapshot",
            "tags": [
                {
                    "key": "no_delete"
                }
            ],
            "state": "available",
            "fromAttachedDisks": [],
            "fromInstanceName": "MEAN-1",
            "fromInstanceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/1761aa0a-6038-4f25-8b94-2EXAMPLE19fd",
            "fromBlueprintId": "wordpress",
            "fromBundleId": "micro_3_0",
            "isFromAutoSnapshot": false,
            "sizeInGb": 40
        },
        {
            "name": "MEAN-1-1571419854",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
            "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
            "createdAt": 1571419891.927,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "InstanceSnapshot",
            "tags": [],
            "state": "available",
            "fromAttachedDisks": [],
            "fromInstanceName": "MEAN-1",
            "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
            "fromBlueprintId": "mean",
            "fromBundleId": "medium_3_0",
            "isFromAutoSnapshot": false,
            "sizeInGb": 80
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstanceSnapshotsà la section Référence des AWS CLI commandes.

`get-instance-state`

L'exemple de code suivant montre comment utiliser`get-instance-state`.

AWS CLI

Pour obtenir des informations sur l'état d'une instance

L'`get-instance-state`exemple suivant renvoie l'état de l'instance spécifiée.

```
aws lightsail get-instance-state \
    --instance-name MEAN-1
```

Sortie :

```
{
    "state": {
        "code": 16,
        "name": "running"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstanceStateà la section Référence des AWS CLI commandes.

`get-instance`

L'exemple de code suivant montre comment utiliser`get-instance`.

AWS CLI

Pour obtenir des informations sur une instance

L'`get-instance`exemple suivant affiche les détails de l'instance`MEAN-1`.

```
aws lightsail get-instance \
    --instance-name MEAN-1
```

Sortie :


```
{
    "instance": {
        "name": "MEAN-1",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-EXAMPLE4bada",
        "supportCode": "6EXAMPLE3362/i-05EXAMPLE407c97d3",
        "createdAt": 1570635023.124,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Instance",
        "tags": [],
        "blueprintId": "mean",
        "blueprintName": "MEAN",
        "bundleId": "medium_3_0",
        "isStaticIp": false,
        "privateIpAddress": "192.0.2.0",
        "publicIpAddress": "192.0.2.0",
        "hardware": {
            "cpuCount": 2,
            "disks": [
                {
                    "createdAt": 1570635023.124,
                    "sizeInGb": 80,
                    "isSystemDisk": true,
                    "iops": 240,
                    "path": "/dev/xvda",
                    "attachedTo": "MEAN-1",
                    "attachmentState": "attached"
                }
            ],
            "ramSizeInGb": 4.0
        },
        "networking": {
            "monthlyTransfer": {
                "gbPerMonthAllocated": 4096
            },
```

```
            "ports": [
                {
                    "fromPort": 80,
                    "toPort": 80,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                },
                {
                    "fromPort": 22,
                    "toPort": 22,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                },
                {
                    "fromPort": 443,
                    "toPort": 443,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                }
            ]
        },
        "state": {
            "code": 16,
            "name": "running"
        },
        "username": "bitnami",
        "sshKeyName": "MyKey"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstanceà la section Référence des AWS CLI commandes.

`get-instances`

L'exemple de code suivant montre comment utiliser`get-instances`.

AWS CLI

Pour obtenir des informations sur toutes les instances

L'`get-instances`exemple suivant affiche les détails de toutes les instances de la AWS région configurée.

```
aws lightsail get-instances
```

Sortie :

```
{
    "instances": [
        {
            "name": "Windows_Server_2022-1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/0f44fbb9-8f55-4e47-a25e-EXAMPLE04763",
            "supportCode": "62EXAMPLE362/i-0bEXAMPLE71a686b9",
            "createdAt": 1571332358.665,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Instance",
            "tags": [],
            "blueprintId": "windows_server_2022",
            "blueprintName": "Windows Server 2022",
            "bundleId": "large_win_3_0",
            "isStaticIp": false,
            "privateIpAddress": "192.0.2.0",
            "publicIpAddress": "192.0.2.0",
            "hardware": {
                "cpuCount": 1,
                "disks": [
                    {
                        "createdAt": 1571332358.665,
                        "sizeInGb": 160,
                        "isSystemDisk": true,
                        "iops": 180,
```

```
                        "path": "/dev/sda1",
                        "attachedTo": "Windows_Server_2022-1",
                        "attachmentState": "attached"
                    },
                    {
                        "name": "my-disk-for-windows-server",
                        "arn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/4123a81c-484c-49ea-afea-5EXAMPLEda87",
                        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEb2b99ca3d",
                        "createdAt": 1571355063.494,
                        "location": {
                            "availabilityZone": "us-west-2a",
                            "regionName": "us-west-2"
                        },
                        "resourceType": "Disk",
                        "tags": [],
                        "sizeInGb": 128,
                        "isSystemDisk": false,
                        "iops": 384,
                        "path": "/dev/xvdf",
                        "state": "in-use",
                        "attachedTo": "Windows_Server_2022-1",
                        "isAttached": true,
                        "attachmentState": "attached"
                    }
                ],
                "ramSizeInGb": 8.0
            },
            "networking": {
                "monthlyTransfer": {
                    "gbPerMonthAllocated": 3072
                },
                "ports": [
                    {
                        "fromPort": 80,
                        "toPort": 80,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 22,
```

```
                        "toPort": 22,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 3389,
                        "toPort": 3389,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    }
                ]
            },
            "state": {
                "code": 16,
                "name": "running"
            },
            "username": "Administrator",
            "sshKeyName": "LightsailDefaultKeyPair"
        },
        {
            "name": "MEAN-1",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-8EXAMPLEbada",
            "supportCode": "6EXAMPLE3362/i-0EXAMPLEa407c97d3",
            "createdAt": 1570635023.124,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Instance",
            "tags": [],
            "blueprintId": "mean",
            "blueprintName": "MEAN",
            "bundleId": "medium_3_0",
            "isStaticIp": false,
            "privateIpAddress": "192.0.2.0",
            "publicIpAddress": "192.0.2.0",
            "hardware": {
```

```
                "cpuCount": 2,
                "disks": [
                    {
                        "name": "Disk-1",
                        "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
                        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
                        "createdAt": 1566585439.587,
                        "location": {
                            "availabilityZone": "us-west-2a",
                            "regionName": "us-west-2"
                        },
                        "resourceType": "Disk",
                        "tags": [
                            {
                                "key": "test"
                            }
                        ],
                        "sizeInGb": 8,
                        "isSystemDisk": false,
                        "iops": 240,
                        "path": "/dev/xvdf",
                        "state": "in-use",
                        "attachedTo": "MEAN-1",
                        "isAttached": true,
                        "attachmentState": "attached"
                    },
                    {
                        "createdAt": 1570635023.124,
                        "sizeInGb": 80,
                        "isSystemDisk": true,
                        "iops": 240,
                        "path": "/dev/sda1",
                        "attachedTo": "MEAN-1",
                        "attachmentState": "attached"
                    }
                ],
                "ramSizeInGb": 4.0
            },
            "networking": {
                "monthlyTransfer": {
                    "gbPerMonthAllocated": 4096
                },
                "ports": [
```

```
                    {
                        "fromPort": 80,
                        "toPort": 80,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 22,
                        "toPort": 22,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 443,
                        "toPort": 443,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    }
                ]
            },
            "state": {
                "code": 16,
                "name": "running"
            },
            "username": "bitnami",
            "sshKeyName": "MyTestKey"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetInstancesà la section Référence des AWS CLI commandes.

`get-key-pair`

L'exemple de code suivant montre comment utiliser`get-key-pair`.

AWS CLI

Pour obtenir des informations sur une paire de clés

L'`get-key-pair`exemple suivant affiche les détails de la paire de clés spécifiée.

```
aws lightsail get-key-pair \
    --key-pair-name MyKey1
```

Sortie :

```
{
    "keyPair": {
        "name": "MyKey1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",
        "supportCode": "6EXAMPLE3362/MyKey1",
        "createdAt": 1571255026.975,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "KeyPair",
        "tags": [],
        "fingerprint": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetKeyPairà la section Référence des AWS CLI commandes.

`get-key-pairs`

L'exemple de code suivant montre comment utiliser`get-key-pairs`.

AWS CLI

Pour obtenir des informations sur toutes les paires de clés

L'`get-key-pairs`exemple suivant affiche les détails de toutes les paires de clés de la AWS région configurée.

```
aws lightsail get-key-pairs
```

Sortie :

```
{
    "keyPairs": [
        {
            "name": "MyKey1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",
            "supportCode": "6EXAMPLE3362/MyKey1",
            "createdAt": 1571255026.975,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "KeyPair",
            "tags": [],
            "fingerprint":
 "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetKeyPairsà la section Référence des AWS CLI commandes.

`get-load-balancer-tls-certificates`

L'exemple de code suivant montre comment utiliser`get-load-balancer-tls-certificates`.

AWS CLI

Pour obtenir des informations sur les certificats TLS pour un équilibreur de charge

L'`get-load-balancer-tls-certificates`exemple suivant affiche des détails sur les certificats TLS pour l'équilibreur de charge spécifié.

```
aws lightsail get-load-balancer-tls-certificates \
```

```
    --load-balancer-name LoadBalancer-1
```

Sortie :

```
{
    "tlsCertificates": [
        {
            "name": "example-com",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancerTlsCertificate/d7bf4643-6a02-4cd4-b3c4-
fEXAMPLE9b4d",
            "supportCode": "6EXAMPLE3362/arn:aws:acm:us-
west-2:333322221111:certificate/9af8e32c-a54e-4a67-8c63-cEXAMPLEb314",
            "createdAt": 1571678025.3,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "LoadBalancerTlsCertificate",
            "loadBalancerName": "LoadBalancer-1",
            "isAttached": false,
            "status": "ISSUED",
            "domainName": "example.com",
            "domainValidationRecords": [
                {
                    "name": "_dEXAMPLE4ede046a0319eb44a4eb3cbc.example.com.",
                    "type": "CNAME",
                    "value": "_bEXAMPLE0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws.",
                    "validationStatus": "SUCCESS",
                    "domainName": "example.com"
                }
            ],
            "issuedAt": 1571678070.0,
            "issuer": "Amazon",
            "keyAlgorithm": "RSA-2048",
            "notAfter": 1605960000.0,
            "notBefore": 1571616000.0,
            "serial": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff",
            "signatureAlgorithm": "SHA256WITHRSA",
            "subject": "CN=example.com",
            "subjectAlternativeNames": [
                "example.com"
```

```
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLoadBalancerTlsCertificatesà la section Référence des AWS CLI commandes.

`get-load-balancer`

L'exemple de code suivant montre comment utiliser`get-load-balancer`.

AWS CLI

Pour obtenir des informations sur un équilibreur de charge

L'`get-load-balancer`exemple suivant affiche des détails sur l'équilibreur de charge spécifié.

```
aws lightsail get-load-balancer \
    --load-balancer-name LoadBalancer-1
```

Sortie :

```
{
    "loadBalancer": {
        "name": "LoadBalancer-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
        "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-
west-2:333322221111:loadbalancer/app/
bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
        "createdAt": 1571677906.723,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "LoadBalancer",
        "tags": [],
        "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-
west-2.elb.amazonaws.com",
        "state": "active",
```

```
        "protocol": "HTTP",
        "publicPorts": [
            80
        ],
        "healthCheckPath": "/",
        "instancePort": 80,
        "instanceHealthSummary": [
            {
                "instanceName": "MEAN-3",
                "instanceHealth": "healthy"
            },
            {
                "instanceName": "MEAN-1",
                "instanceHealth": "healthy"
            },
            {
                "instanceName": "MEAN-2",
                "instanceHealth": "healthy"
            }
        ],
        "tlsCertificateSummaries": [
            {
                "name": "example-com",
                "isAttached": false
            }
        ],
        "configurationOptions": {
            "SessionStickinessEnabled": "false",
            "SessionStickiness_LB_CookieDurationSeconds": "86400"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetLoadBalancerà la section Référence des AWS CLI commandes.

`get-load-balancers`

L'exemple de code suivant montre comment utiliser`get-load-balancers`.

AWS CLI

Pour obtenir des informations sur tous les équilibreurs de charge

L'`get-load-balancers`exemple suivant affiche les détails de tous les équilibreurs de charge de la AWS région configurée.

```
aws lightsail get-load-balancers
```

Sortie :

```
{
    "loadBalancers": [
        {
            "name": "LoadBalancer-1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
            "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-
west-2:333322221111:loadbalancer/app/
bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
            "createdAt": 1571677906.723,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "LoadBalancer",
            "tags": [],
            "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-
west-2.elb.amazonaws.com",
            "state": "active",
            "protocol": "HTTP",
            "publicPorts": [
                80
            ],
            "healthCheckPath": "/",
            "instancePort": 80,
            "instanceHealthSummary": [
                {
                    "instanceName": "MEAN-3",
                    "instanceHealth": "healthy"
                },
                {
                    "instanceName": "MEAN-1",
                    "instanceHealth": "healthy"
                },
                {
                    "instanceName": "MEAN-2",
```

```
                    "instanceHealth": "healthy"
                }
            ],
            "tlsCertificateSummaries": [
                {
                    "name": "example-com",
                    "isAttached": false
                }
            ],
            "configurationOptions": {
                "SessionStickinessEnabled": "false",
                "SessionStickiness_LB_CookieDurationSeconds": "86400"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLoadBalancersà la section Référence des AWS CLI commandes.

`get-operation`

L'exemple de code suivant montre comment utiliser`get-operation`.

AWS CLI

Pour obtenir des informations sur une seule opération

L'`get-operation`exemple suivant affiche les détails de l'opération spécifiée.

```
aws lightsail get-operation \
    --operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

Sortie :

```
{
    "operation": {
        "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
        "resourceName": "Instance-1",
        "resourceType": "Instance",
        "createdAt": 1571679872.404,
```

```
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationType": "CreateInstance",
        "status": "Succeeded",
        "statusChangedAt": 1571679890.304
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetOperationà la section Référence des AWS CLI commandes.

`get-operations-for-resource`

L'exemple de code suivant montre comment utiliser`get-operations-for-resource`.

AWS CLI

Pour obtenir toutes les opérations relatives à une ressource

L'`get-operations-for-resource`exemple suivant affiche les détails de toutes les opérations pour la ressource spécifiée.

```
aws lightsail get-operations-for-resource \
    --resource-name LoadBalancer-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678786.071,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "DetachInstancesFromLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571679087.57
        },
        {
            "id": "2d742a18-0e7f-48c8-9705-3EXAMPLEf98a",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678782.784,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571678798.465
        },
        {
            "id": "6c700fcc-4246-40ab-952b-1EXAMPLEdac2",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678775.297,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-3",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571678842.806
        },
        ...
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous GetOperationsForResourceà la section Référence des AWS CLI commandes.

`get-operations`

L'exemple de code suivant montre comment utiliser`get-operations`.

AWS CLI

Pour obtenir des informations sur toutes les opérations

L'`get-operations`exemple suivant affiche les détails de toutes les opérations dans la AWS région configurée.

```
aws lightsail get-operations
```

Sortie :

```
{
    "operations": [
        {
            "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
            "resourceName": "Instance-1",
            "resourceType": "Instance",
            "createdAt": 1571679872.404,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "CreateInstance",
            "status": "Succeeded",
            "statusChangedAt": 1571679890.304
        },
        {
            "id": "701a3339-930e-4914-a9f9-7EXAMPLE68d7",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571678786.072,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "LoadBalancer-1",
            "operationType": "DetachInstancesFromLoadBalancer",
```

```
            "status": "Succeeded",
            "statusChangedAt": 1571679086.399
        },
        {
            "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678786.071,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "WordPress-1",
            "operationType": "DetachInstancesFromLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571679087.57
        },
        ...
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetOperationsà la section Référence des AWS CLI commandes.

`get-regions`

L'exemple de code suivant montre comment utiliser`get-regions`.

AWS CLI

Pour accéder à toutes les AWS régions pour Amazon Lightsail

L'`get-regions`exemple suivant affiche des informations détaillées sur toutes les AWS régions pour Amazon Lightsail.

```
aws lightsail get-regions
```

Sortie :

```
{
```

```
    "regions": [
        {
            "continentCode": "NA",
            "description": "This region is recommended to serve users in the eastern
 United States",
            "displayName": "Virginia",
            "name": "us-east-1",
            "availabilityZones": [],
            "relationalDatabaseAvailabilityZones": []
        },
        {
            "continentCode": "NA",
            "description": "This region is recommended to serve users in the eastern
 United States",
            "displayName": "Ohio",
            "name": "us-east-2",
            "availabilityZones": [],
            "relationalDatabaseAvailabilityZones": []
        },
        {
            "continentCode": "NA",
            "description": "This region is recommended to serve users in the
 northwestern United States, Alaska, and western Canada",
            "displayName": "Oregon",
            "name": "us-west-2",
            "availabilityZones": [],
            "relationalDatabaseAvailabilityZones": []
        },
        ...
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRegionsà la section Référence des AWS CLI commandes.

`get-relational-database-blueprints`

L'exemple de code suivant montre comment utiliser`get-relational-database-blueprints`.

AWS CLI

Pour obtenir les plans des nouvelles bases de données relationnelles

L'`get-relational-database-blueprints`exemple suivant affiche des détails sur tous les plans de base de données relationnelle disponibles qui peuvent être utilisés pour créer de nouvelles bases de données relationnelles dans Amazon Lightsail.

```
aws lightsail get-relational-database-blueprints
```

Sortie :

```
{
    "blueprints": [
        {
            "blueprintId": "mysql_5_6",
            "engine": "mysql",
            "engineVersion": "5.6.44",
            "engineDescription": "MySQL Community Edition",
            "engineVersionDescription": "MySQL 5.6.44",
            "isEngineDefault": false
        },
        {
            "blueprintId": "mysql_5_7",
            "engine": "mysql",
            "engineVersion": "5.7.26",
            "engineDescription": "MySQL Community Edition",
            "engineVersionDescription": "MySQL 5.7.26",
            "isEngineDefault": true
        },
        {
            "blueprintId": "mysql_8_0",
            "engine": "mysql",
            "engineVersion": "8.0.16",
            "engineDescription": "MySQL Community Edition",
            "engineVersionDescription": "MySQL 8.0.16",
            "isEngineDefault": false
        },
        {
            "blueprintId": "postgres_9_6",
            "engine": "postgres",
            "engineVersion": "9.6.15",
            "engineDescription": "PostgreSQL",
            "engineVersionDescription": "PostgreSQL 9.6.15-R1",
            "isEngineDefault": false
        },
        {
```

```
            "blueprintId": "postgres_10",
            "engine": "postgres",
            "engineVersion": "10.10",
            "engineDescription": "PostgreSQL",
            "engineVersionDescription": "PostgreSQL 10.10-R1",
            "isEngineDefault": false
        },
        {
            "blueprintId": "postgres_11",
            "engine": "postgres",
            "engineVersion": "11.5",
            "engineDescription": "PostgreSQL",
            "engineVersionDescription": "PostgreSQL 11.5-R1",
            "isEngineDefault": true
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseBlueprintsà la section Référence des AWS CLI commandes.

`get-relational-database-bundles`

L'exemple de code suivant montre comment utiliser`get-relational-database-bundles`.

AWS CLI

Pour obtenir les ensembles pour les nouvelles bases de données relationnelles

L'`get-relational-database-bundles`exemple suivant affiche des informations détaillées sur tous les ensembles de bases de données relationnelles disponibles qui peuvent être utilisés pour créer de nouvelles bases de données relationnelles dans Amazon Lightsail. Notez que la réponse n'inclut pas les paquets inactifs car l'`--include-inactive`indicateur n'est pas spécifié dans la commande. Vous ne pouvez pas utiliser de bundles inactifs pour créer de nouvelles bases de données relationnelles.

```
aws lightsail get-relational-database-bundles
```

Sortie :

```
{
```

```
    "bundles": [
        {
            "bundleId": "micro_2_0",
            "name": "Micro",
            "price": 15.0,
            "ramSizeInGb": 1.0,
            "diskSizeInGb": 40,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "micro_ha_2_0",
            "name": "Micro with High Availability",
            "price": 30.0,
            "ramSizeInGb": 1.0,
            "diskSizeInGb": 40,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "small_2_0",
            "name": "Small",
            "price": 30.0,
            "ramSizeInGb": 2.0,
            "diskSizeInGb": 80,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "small_ha_2_0",
            "name": "Small with High Availability",
            "price": 60.0,
            "ramSizeInGb": 2.0,
            "diskSizeInGb": 80,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
```

```
        },
        {
            "bundleId": "medium_2_0",
            "name": "Medium",
            "price": 60.0,
            "ramSizeInGb": 4.0,
            "diskSizeInGb": 120,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "medium_ha_2_0",
            "name": "Medium with High Availability",
            "price": 120.0,
            "ramSizeInGb": 4.0,
            "diskSizeInGb": 120,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "large_2_0",
            "name": "Large",
            "price": 115.0,
            "ramSizeInGb": 8.0,
            "diskSizeInGb": 240,
            "transferPerMonthInGb": 200,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "large_ha_2_0",
            "name": "Large with High Availability",
            "price": 230.0,
            "ramSizeInGb": 8.0,
            "diskSizeInGb": 240,
            "transferPerMonthInGb": 200,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Création d'une base de données dans Amazon Lightsail dans le manuel Amazon Lightsail Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseBundlesà la section Référence des AWS CLI commandes.

`get-relational-database-events`

L'exemple de code suivant montre comment utiliser`get-relational-database-events`.

AWS CLI

Pour obtenir les événements d'une base de données relationnelle

L'`get-relational-database-events`exemple suivant affiche les détails des événements survenus au cours des 17 dernières heures (10 à 20 minutes) pour la base de données relationnelle spécifiée.

```
aws lightsail get-relational-database-events \
    --relational-database-name Database-1 \
    --duration-in-minutes 1020
```

Sortie :

```
{
    "relationalDatabaseEvents": [
        {
            "resource": "Database-1",
            "createdAt": 1571654146.553,
            "message": "Backing up Relational Database",
            "eventCategories": [
                "backup"
            ]
        },
        {
            "resource": "Database-1",
            "createdAt": 1571654249.98,
            "message": "Finished Relational Database backup",
```

```
            "eventCategories": [
                "backup"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseEventsà la section Référence des AWS CLI commandes.

`get-relational-database-log-events`

L'exemple de code suivant montre comment utiliser`get-relational-database-log-events`.

AWS CLI

Pour obtenir les événements du journal d'une base de données relationnelle

L'`get-relational-database-log-events`exemple suivant affiche des détails sur le journal spécifié entre `1570733176` et `1571597176` pour la base `Database1` de données relationnelle. Les informations renvoyées sont configurées pour commencer à partir de`head`.

Nous vous recommandons d'utiliser un convertisseur de temps Unix pour identifier les heures de début et de fin.

```
aws lightsail get-relational-database-log-events \
    --relational-database-name Database1 \
    --log-stream-name error \
    --start-from-head \
    --start-time 1570733176 \
    --end-time 1571597176
```

Sortie :

```
{
    "resourceLogEvents": [
        {
            "createdAt": 1570820267.0,
            "message": "2019-10-11 18:57:47 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Name or service not known"
        },
        {
```

```
            "createdAt": 1570860974.0,
            "message": "2019-10-12 06:16:14 20969 [Warning] IP address '8192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860977.0,
            "message": "2019-10-12 06:16:17 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860979.0,
            "message": "2019-10-12 06:16:19 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860981.0,
            "message": "2019-10-12 06:16:21 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860982.0,
            "message": "2019-10-12 06:16:22 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860984.0,
            "message": "2019-10-12 06:16:24 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860986.0,
            "message": "2019-10-12 06:16:26 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        ...
        }
    ],
    "nextBackwardToken":
 "eEXAMPLEZXJUZXh0IjoiZnRWb3F3cUpRSlQ5NndMYThxelRUZlFhR3J6c2dKWEEvM2kvajZMZzVVVWpqRDN0YjFXTj
    "nextForwardToken":
 "eEXAMPLEZXJUZXh0IjoiT09Lb0Z6ZFRJbHhaNEQ5N2tPbkkwRmwwNUxPZjFTbFFwUklQbzlSaWgvMWVXbEk4aG56VH
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseLogEventsà la section Référence des AWS CLI commandes.

`get-relational-database-log-streams`

L'exemple de code suivant montre comment utiliser`get-relational-database-log-streams`.

AWS CLI

Pour obtenir les flux de log d'une base de données relationnelle

L'`get-relational-database-log-streams`exemple suivant renvoie tous les flux de journaux disponibles pour la base de données relationnelle spécifiée.

```
aws lightsail get-relational-database-log-streams \
--relational-database-name Database1
```

Sortie :

```
{
    "logStreams": [
        "audit",
        "error",
        "general",
        "slowquery"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseLogStreamsà la section Référence des AWS CLI commandes.

`get-relational-database-master-user-password`

L'exemple de code suivant montre comment utiliser`get-relational-database-master-user-password`.

AWS CLI

Pour obtenir le mot de passe de l'utilisateur principal pour une base de données relationnelle

L'`get-relational-database-master-user-password`exemple suivant renvoie des informations sur le mot de passe de l'utilisateur principal pour la base de données relationnelle spécifiée.

```
aws lightsail get-relational-database-master-user-password \
    --relational-database-name Database-1
```

Sortie :

```
{
    "masterUserPassword": "VEXAMPLEec.9qvx,_t<)Wkf)kwboM,>2",
    "createdAt": 1571259453.959
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseMasterUserPasswordà la section Référence des AWS CLI commandes.

`get-relational-database-metric-data`

L'exemple de code suivant montre comment utiliser`get-relational-database-metric-data`.

AWS CLI

Pour obtenir des données métriques pour une base de données relationnelle

L'`get-relational-database-metric-data`exemple suivant renvoie le décompte de la métrique `DatabaseConnections` sur la période de 24 heures (`86400`secondes) entre `1570733176` et `1571597176` pour la base `Database1` de données relationnelle.

Nous vous recommandons d'utiliser un convertisseur de temps Unix pour identifier les heures de début et de fin.

```
aws lightsail get-relational-database-metric-data \
    --relational-database-name Database1 \
    --metric-name DatabaseConnections \
    --period 86400 \
    --start-time 1570733176 \
    --end-time 1571597176 \
    --unit Count \
    --statistics Sum
```

Sortie :


```
{
    "metricName": "DatabaseConnections",
    "metricData": [
        {
            "sum": 1.0,
            "timestamp": 1571510760.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1570733160.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1570992360.0,
            "unit": "Count"
        },
        {
            "sum": 0.0,
            "timestamp": 1571251560.0,
            "unit": "Count"
        },
        {
            "sum": 721.0,
            "timestamp": 1570819560.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1571078760.0,
            "unit": "Count"
        },
        {
            "sum": 2.0,
            "timestamp": 1571337960.0,
            "unit": "Count"
        },
        {
            "sum": 684.0,
            "timestamp": 1570905960.0,
            "unit": "Count"
```

```
        },
        {
            "sum": 0.0,
            "timestamp": 1571165160.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1571424360.0,
            "unit": "Count"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseMetricDataà la section Référence des AWS CLI commandes.

`get-relational-database-parameters`

L'exemple de code suivant montre comment utiliser`get-relational-database-parameters`.

AWS CLI

Pour obtenir les paramètres d'une base de données relationnelle

L'`get-relational-database-parameters`exemple suivant renvoie des informations sur tous les paramètres disponibles pour la base de données relationnelle spécifiée.

```
aws lightsail get-relational-database-parameters \
    --relational-database-name Database-1
```

Sortie :

```
{
    "parameters": [
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "dynamic",
            "dataType": "boolean",
            "description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
```

```
            "isModifiable": true,
            "parameterName": "activate_all_roles_on_login",
            "parameterValue": "0"
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "static",
            "dataType": "boolean",
            "description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
            "isModifiable": false,
            "parameterName": "allow-suspicious-udfs"
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "dynamic",
            "dataType": "boolean",
            "description": "Sets the autocommit mode",
            "isModifiable": true,
            "parameterName": "autocommit"
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "static",
            "dataType": "boolean",
            "description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "isModifiable": false,
            "parameterName": "auto_generate_certs"
        },
        ...
        }
    ]
}
```

Pour plus d'informations, consultez la section Mise à jour des paramètres de base de données dans Amazon Lightsail dans le Guide de développement de Lightsail.

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseParametersà la section Référence des AWS CLI commandes.

`get-relational-database-snapshot`

L'exemple de code suivant montre comment utiliser`get-relational-database-snapshot`.

AWS CLI

Pour obtenir des informations sur un instantané de base de données relationnelle

L'`get-relational-database-snapshot`exemple suivant affiche les détails relatifs à l'instantané de base de données relationnelle spécifié.

```
aws lightsail get-relational-database-snapshot \
    --relational-database-snapshot-name Database-1-1571350042
```

Sortie :

```
{
    "relationalDatabaseSnapshot": {
        "name": "Database-1-1571350042",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9EXAMPLEaee3643d2",
        "supportCode": "6EXAMPLE3362/ls-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
        "createdAt": 1571350046.238,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "RelationalDatabaseSnapshot",
        "tags": [],
        "engine": "mysql",
        "engineVersion": "8.0.16",
        "sizeInGb": 40,
        "state": "available",
        "fromRelationalDatabaseName": "Database-1",
        "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
        "fromRelationalDatabaseBundleId": "micro_1_0",
        "fromRelationalDatabaseBlueprintId": "mysql_8_0"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseSnapshotà la section Référence des AWS CLI commandes.

`get-relational-database-snapshots`

L'exemple de code suivant montre comment utiliser`get-relational-database-snapshots`.

AWS CLI

Pour obtenir des informations sur tous les instantanés de base de données relationnelle

L'`get-relational-database-snapshots`exemple suivant affiche des détails sur tous les instantanés de base de données relationnelle dans la région configurée AWS .

```
aws lightsail get-relational-database-snapshots
```

Sortie :

```
{
    "relationalDatabaseSnapshots": [
        {
            "name": "Database-1-1571350042",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9861-6EXAMPLE43d2",
            "supportCode": "6EXAMPLE3362/
ls-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
            "createdAt": 1571350046.238,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabaseSnapshot",
            "tags": [],
            "engine": "mysql",
            "engineVersion": "8.0.16",
            "sizeInGb": 40,
            "state": "available",
            "fromRelationalDatabaseName": "Database-1",
            "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
            "fromRelationalDatabaseBundleId": "micro_1_0",
            "fromRelationalDatabaseBlueprintId": "mysql_8_0"
        },
        {
            "name": "Database1-Console",
```

```
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/8b94136e-06ec-4b1a-
a3fb-5EXAMPLEe1e9",
            "supportCode": "6EXAMPLE3362/
ls-9EXAMPLE14b000d34c8d1c432734e137612d5b5c",
            "createdAt": 1571249981.025,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabaseSnapshot",
            "tags": [
                {
                    "key": "test"
                }
            ],
            "engine": "mysql",
            "engineVersion": "5.6.44",
            "sizeInGb": 40,
            "state": "available",
            "fromRelationalDatabaseName": "Database1",
            "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/a6161cb7-4535-4f16-9dcf-8EXAMPLE3d4e",
            "fromRelationalDatabaseBundleId": "micro_1_0",
            "fromRelationalDatabaseBlueprintId": "mysql_5_6"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseSnapshotsà la section Référence des AWS CLI commandes.

`get-relational-database`

L'exemple de code suivant montre comment utiliser`get-relational-database`.

AWS CLI

Pour obtenir des informations sur une base de données relationnelle

L'`get-relational-database`exemple suivant affiche des détails sur la base de données relationnelle spécifiée.

```
aws lightsail get-relational-database \
    --relational-database-name Database-1
```

Sortie :


```
{
    "relationalDatabase": {
        "name": "Database-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
        "supportCode": "6EXAMPLE3362/ls-9EXAMPLE8ad863723b62cc8901a8aa6e794ae0d2",
        "createdAt": 1571259453.795,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "RelationalDatabase",
        "tags": [],
        "relationalDatabaseBlueprintId": "mysql_8_0",
        "relationalDatabaseBundleId": "micro_1_0",
        "masterDatabaseName": "dbmaster",
        "hardware": {
            "cpuCount": 1,
            "diskSizeInGb": 40,
            "ramSizeInGb": 1.0
        },
        "state": "available",
        "backupRetentionEnabled": false,
        "pendingModifiedValues": {},
        "engine": "mysql",
        "engineVersion": "8.0.16",
        "masterUsername": "dbmasteruser",
        "parameterApplyStatus": "in-sync",
        "preferredBackupWindow": "10:01-10:31",
        "preferredMaintenanceWindow": "sat:11:14-sat:11:44",
        "publiclyAccessible": true,
        "masterEndpoint": {
            "port": 3306,
            "address": "ls-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-
west-2.rds.amazonaws.com"
        },
        "pendingMaintenanceActions": []
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabaseà la section Référence des AWS CLI commandes.

`get-relational-databases`

L'exemple de code suivant montre comment utiliser`get-relational-databases`.

AWS CLI

Pour obtenir des informations sur toutes les bases de données relationnelles

L'`get-relational-databases`exemple suivant affiche les détails de toutes les bases de données relationnelles de la AWS région configurée.

```
aws lightsail get-relational-databases
```

Sortie :

```
{
    "relationalDatabases": [
        {
            "name": "MySQL",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",
            "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEa995d8c3b06b4501356e5f2f28e1aeba",
            "createdAt": 1554306019.155,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabase",
            "tags": [],
            "relationalDatabaseBlueprintId": "mysql_8_0",
            "relationalDatabaseBundleId": "micro_1_0",
            "masterDatabaseName": "dbmaster",
            "hardware": {
                "cpuCount": 1,
                "diskSizeInGb": 40,
                "ramSizeInGb": 1.0
```

```
            },
            "state": "available",
            "backupRetentionEnabled": true,
            "pendingModifiedValues": {},
            "engine": "mysql",
            "engineVersion": "8.0.15",
            "latestRestorableTime": 1571686200.0,
            "masterUsername": "dbmasteruser",
            "parameterApplyStatus": "in-sync",
            "preferredBackupWindow": "07:51-08:21",
            "preferredMaintenanceWindow": "tue:12:18-tue:12:48",
            "publiclyAccessible": true,
            "masterEndpoint": {
                "port": 3306,
                "address":
 "ls-3EXAMPLEa995d8c3b06b4501356e5f2fEXAMPLEa.czowadgeezqi.us-
west-2.rds.amazonaws.com"
            },
            "pendingMaintenanceActions": []
        },
        {
            "name": "Postgres",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/
e9780b6b-d0ab-4af2-85f1-1EXAMPLEac68",
            "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEb4fffb5cec056220c734713e14bd5fcd",
            "createdAt": 1554306000.814,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabase",
            "tags": [],
            "relationalDatabaseBlueprintId": "postgres_11",
            "relationalDatabaseBundleId": "micro_1_0",
            "masterDatabaseName": "dbmaster",
            "hardware": {
                "cpuCount": 1,
                "diskSizeInGb": 40,
                "ramSizeInGb": 1.0
            },
            "state": "available",
            "backupRetentionEnabled": true,
            "pendingModifiedValues": {},
```

```
            "engine": "postgres",
            "engineVersion": "11.1",
            "latestRestorableTime": 1571686339.0,
            "masterUsername": "dbmasteruser",
            "parameterApplyStatus": "in-sync",
            "preferredBackupWindow": "06:19-06:49",
            "preferredMaintenanceWindow": "sun:10:19-sun:10:49",
            "publiclyAccessible": false,
            "masterEndpoint": {
                "port": 5432,
                "address":
 "ls-3EXAMPLEb4fffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
            },
            "pendingMaintenanceActions": []
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetRelationalDatabasesà la section Référence des AWS CLI commandes.

`get-static-ip`

L'exemple de code suivant montre comment utiliser`get-static-ip`.

AWS CLI

Pour obtenir des informations sur une adresse IP statique

L'`get-static-ip`exemple suivant affiche les détails relatifs à l'adresse IP statique spécifiée.

```
aws lightsail get-static-ip \
    --static-ip-name StaticIp-1
```

Sortie :

```
{
    "staticIp": {
        "name": "StaticIp-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-82e2-2EXAMPLE23ad",
```

```
        "supportCode": "6EXAMPLE3362/192.0.2.0",
        "createdAt": 1571071325.076,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "StaticIp",
        "ipAddress": "192.0.2.0",
        "isAttached": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetStaticIpà la section Référence des AWS CLI commandes.

`get-static-ips`

L'exemple de code suivant montre comment utiliser`get-static-ips`.

AWS CLI

Pour obtenir des informations sur toutes les statiques IPs

L'`get-static-ips`exemple suivant affiche des détails sur tous les éléments statiques IPs de la AWS région configurée.

```
aws lightsail get-static-ips
```

Sortie :

```
{
    "staticIps": [
        {
            "name": "StaticIp-1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-8EXAMPLE16f9423ad",
            "supportCode": "6EXAMPLE3362/192.0.2.0",
            "createdAt": 1571071325.076,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "resourceType": "StaticIp",
            "ipAddress": "192.0.2.0",
            "isAttached": false
        },
        {
            "name": "StaticIP-2",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:StaticIp/c61edb40-
e5f0-4fd6-ae7c-8EXAMPLE19f8",
            "supportCode": "6EXAMPLE3362/192.0.2.2",
            "createdAt": 1568305385.681,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "StaticIp",
            "ipAddress": "192.0.2.2",
            "attachedTo": "WordPress-1",
            "isAttached": true
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetStaticIpsà la section Référence des AWS CLI commandes.

`is-vpc-peered`

L'exemple de code suivant montre comment utiliser`is-vpc-peered`.

AWS CLI

Pour déterminer si votre cloud privé virtuel Amazon Lightsail est pair

L'`is-vpc-peered`exemple suivant renvoie l'état de peering du cloud privé virtuel (VPC) Amazon Lightsail pour la région spécifiée. AWS

```
aws lightsail is-vpc-peered \
    --region us-west-2
```

Sortie :

```
{
```

```
    "isPeered": true
}
```

- Pour plus de détails sur l'API, reportez-vous IsVpcPeeredà la section Référence des AWS CLI commandes.

`open-instance-public-ports`

L'exemple de code suivant montre comment utiliser`open-instance-public-ports`.

AWS CLI

Pour ouvrir des ports de pare-feu pour une instance

L'`open-instance-public-ports`exemple suivant ouvre le port TCP 22 sur l'instance spécifiée.

```
aws lightsail open-instance-public-ports \
    --instance-name MEAN-2 \
    --port-info fromPort=22,protocol=TCP,toPort=22
```

Sortie :

```
{
    "operation": {
        "id": "719744f0-a022-46f2-9f11-6EXAMPLE4642",
        "resourceName": "MEAN-2",
        "resourceType": "Instance",
        "createdAt": 1571072906.849,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "22/tcp",
        "operationType": "OpenInstancePublicPorts",
        "status": "Succeeded",
        "statusChangedAt": 1571072906.849
    }
}
```

- Pour plus de détails sur l'API, reportez-vous OpenInstancePublicPortsà la section Référence des AWS CLI commandes.

`peer-vpc`

L'exemple de code suivant montre comment utiliser`peer-vpc`.

AWS CLI

Pour comparer le cloud privé virtuel Amazon Lightsail

L'`peer-vpc`exemple suivant compare le cloud privé virtuel (VPC) Amazon Lightsail pour la région spécifiée. AWS

```
aws lightsail peer-vpc \
    --region us-west-2
```

Sortie :

```
{
    "operation": {
        "id": "787e846a-54ac-497f-bce2-9EXAMPLE5d91",
        "resourceName": "vpc-0EXAMPLEa5261efb3",
        "resourceType": "PeeredVpc",
        "createdAt": 1571694233.104,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "vpc-e2b3eb9b",
        "operationType": "PeeredVpc",
        "status": "Succeeded",
        "statusChangedAt": 1571694233.104
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PeerVpcà la section Référence des AWS CLI commandes.

`reboot-instance`

L'exemple de code suivant montre comment utiliser`reboot-instance`.

AWS CLI

Pour redémarrer une instance

L'`reboot-instance`exemple suivant redémarre l'instance spécifiée.

```
aws lightsail reboot-instance \
    --instance-name MEAN-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "2b679f1c-8b71-4bb4-8e97-8EXAMPLEed93",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571694445.49,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "",
            "operationType": "RebootInstance",
            "status": "Succeeded",
            "statusChangedAt": 1571694445.49
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous RebootInstanceà la section Référence des AWS CLI commandes.

`reboot-relational-database`

L'exemple de code suivant montre comment utiliser`reboot-relational-database`.

AWS CLI

Pour redémarrer une base de données relationnelle

L'`reboot-relational-database`exemple suivant redémarre la base de données relationnelle spécifiée.

```
aws lightsail reboot-relational-database \
    --relational-database-name Database-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "e4c980c0-3137-496c-9c91-1EXAMPLEdec2",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1571694532.91,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "",
            "operationType": "RebootRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1571694532.91
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous RebootRelationalDatabaseà la section Référence des AWS CLI commandes.

`release-static-ip`

L'exemple de code suivant montre comment utiliser`release-static-ip`.

AWS CLI

Pour supprimer une adresse IP statique

L'`release-static-ip`exemple suivant supprime l'adresse IP statique spécifiée.

```
aws lightsail release-static-ip \
    --static-ip-name StaticIp-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "e374c002-dc6d-4c7f-919f-2EXAMPLE13ce",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571694962.003,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "ReleaseStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571694962.003
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ReleaseStaticIpà la section Référence des AWS CLI commandes.

`start-instance`

L'exemple de code suivant montre comment utiliser`start-instance`.

AWS CLI

Pour démarrer une instance

L'`start-instance`exemple suivant démarre l'instance spécifiée.

```
aws lightsail start-instance \
    --instance-name WordPress-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "f88d2a93-7cea-4165-afce-2d688cb18f23",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571695583.463,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StartInstance",
            "status": "Started",
            "statusChangedAt": 1571695583.463
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous StartInstanceà la section Référence des AWS CLI commandes.

`start-relational-database`

L'exemple de code suivant montre comment utiliser`start-relational-database`.

AWS CLI

Pour démarrer une base de données relationnelle

L'`start-relational-database`exemple suivant démarre la base de données relationnelle spécifiée.

```
aws lightsail start-relational-database \
    --relational-database-name Database-1
```

Sortie :

```
{
```

```
    "operations": [
        {
            "id": "4d5294ec-a38a-4fda-9e37-aEXAMPLE0d24",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1571695998.822,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StartRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1571695998.822
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous StartRelationalDatabaseà la section Référence des AWS CLI commandes.

`stop-instance`

L'exemple de code suivant montre comment utiliser`stop-instance`.

AWS CLI

Pour arrêter une instance

L'`stop-instance`exemple suivant arrête l'instance spécifiée.

```
aws lightsail stop-instance \
--instance-name WordPress-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "265357e2-2943-4d51-888a-1EXAMPLE7585",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
```

```
            "createdAt": 1571695471.134,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StopInstance",
            "status": "Started",
            "statusChangedAt": 1571695471.134
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous StopInstanceà la section Référence des AWS CLI commandes.

`stop-relational-database`

L'exemple de code suivant montre comment utiliser`stop-relational-database`.

AWS CLI

Pour arrêter une base de données relationnelle

L'`stop-relational-database`exemple suivant arrête la base de données relationnelle spécifiée.

```
aws lightsail stop-relational-database \
    --relational-database-name Database-1
```

Sortie :

```
{
    "operations": [
        {
            "id": "cc559c19-4adb-41e4-b75b-5EXAMPLE4e61",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1571695526.29,
            "location": {
                "availabilityZone": "us-west-2a",
```

```
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StopRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1571695526.29
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous StopRelationalDatabaseà la section Référence des AWS CLI commandes.

`unpeer-vpc`

L'exemple de code suivant montre comment utiliserunpeer-vpc.

AWS CLI

Pour déconnecter le cloud privé virtuel Amazon Lightsail

L'unpeer-vpcexemple suivant annule l'appariement du cloud privé virtuel (VPC) Amazon Lightsail pour la région spécifiée. AWS

```
aws lightsail unpeer-vpc \
    --region us-west-2
```

Sortie :

```
{
    "operation": {
        "id": "531aca64-7157-47ab-84c6-eEXAMPLEd898",
        "resourceName": "vpc-0EXAMPLEa5261efb3",
        "resourceType": "PeeredVpc",
        "createdAt": 1571694109.945,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "vpc-e2b3eb9b",
        "operationType": "UnpeeredVpc",
```

```
        "status": "Succeeded",
        "statusChangedAt": 1571694109.945
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UnpeerVpcà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Macie AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Macie.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-buckets`

L'exemple de code suivant montre comment utiliser`describe-buckets`.

AWS CLI

Pour demander des données concernant un ou plusieurs compartiments S3 surveillés et analysés par Amazon Macie pour votre compte

L'`describe-buckets`exemple suivant interroge les métadonnées de tous les compartiments S3 dont le nom commence par amzn-s3-demo-bucket et qui se trouvent dans la région actuelle. AWS

```
aws macie2 describe-buckets \
    --criteria '{"bucketName":{"prefix":"amzn-s3-demo-bucket"}}'
```

Sortie :


```
{
    "buckets": [
        {
            "accountId": "123456789012",
            "allowsUnencryptedObjectUploads": "FALSE",
            "automatedDiscoveryMonitoringStatus": "MONITORED",
            "bucketArn": "arn:aws:s3:::amzn-s3-demo-bucket1",
            "bucketCreatedAt": "2020-05-18T19:54:00+00:00",
            "bucketName": "amzn-s3-demo-bucket1",
            "classifiableObjectCount": 13,
            "classifiableSizeInBytes": 1592088,
            "jobDetails": {
                "isDefinedInJob": "TRUE",
                "isMonitoredByJob": "TRUE",
                "lastJobId": "08c81dc4a2f3377fae45c9ddaEXAMPLE",
                "lastJobRunTime": "2024-08-19T14:55:30.270000+00:00"
            },
            "lastAutomatedDiscoveryTime": "2024-10-22T19:11:25.364000+00:00",
            "lastUpdated": "2024-10-25T07:33:06.337000+00:00",
            "objectCount": 13,
            "objectCountByEncryptionType": {
                "customerManaged": 0,
                "kmsManaged": 2,
                "s3Managed": 7,
                "unencrypted": 4,
                "unknown": 0
            },
            "publicAccess": {
                "effectivePermission": "NOT_PUBLIC",
                "permissionConfiguration": {
                    "accountLevelPermissions": {
                        "blockPublicAccess": {
                            "blockPublicAcls": true,
                            "blockPublicPolicy": true,
                            "ignorePublicAcls": true,
                            "restrictPublicBuckets": true
                        }
                    },
                    "bucketLevelPermissions": {
                        "accessControlList": {
                            "allowsPublicReadAccess": false,
                            "allowsPublicWriteAccess": false
```

```
                    },
                    "blockPublicAccess": {
                        "blockPublicAcls": true,
                        "blockPublicPolicy": true,
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    },
                    "bucketPolicy": {
                        "allowsPublicReadAccess": false,
                        "allowsPublicWriteAccess": false
                    }
                }
            }
        },
        "region": "us-west-2",
        "replicationDetails": {
            "replicated": false,
            "replicatedExternally": false,
            "replicationAccounts": []
        },
        "sensitivityScore": 78,
        "serverSideEncryption": {
            "kmsMasterKeyId": null,
            "type": "NONE"
        },
        "sharedAccess": "NOT_SHARED",
        "sizeInBytes": 4549746,
        "sizeInBytesCompressed": 0,
        "tags": [
            {
                "key": "Division",
                "value": "HR"
            },
            {
                "key": "Team",
                "value": "Recruiting"
            }
        ],
        "unclassifiableObjectCount": {
            "fileType": 0,
            "storageClass": 0,
            "total": 0
        },
        "unclassifiableObjectSizeInBytes": {
```

```
                "fileType": 0,
                "storageClass": 0,
                "total": 0
            },
            "versioning": true
        },
        {
            "accountId": "123456789012",
            "allowsUnencryptedObjectUploads": "TRUE",
            "automatedDiscoveryMonitoringStatus": "MONITORED",
            "bucketArn": "arn:aws:s3:::amzn-s3-demo-bucket2",
            "bucketCreatedAt": "2020-11-25T18:24:38+00:00",
            "bucketName": "amzn-s3-demo-bucket2",
            "classifiableObjectCount": 8,
            "classifiableSizeInBytes": 133810,
            "jobDetails": {
                "isDefinedInJob": "TRUE",
                "isMonitoredByJob": "FALSE",
                "lastJobId": "188d4f6044d621771ef7d65f2EXAMPLE",
                "lastJobRunTime": "2024-07-09T19:37:11.511000+00:00"
            },
            "lastAutomatedDiscoveryTime": "2024-10-24T19:11:25.364000+00:00",
            "lastUpdated": "2024-10-25T07:33:06.337000+00:00",
            "objectCount": 8,
            "objectCountByEncryptionType": {
                "customerManaged": 0,
                "kmsManaged": 0,
                "s3Managed": 8,
                "unencrypted": 0,
                "unknown": 0
            },
            "publicAccess": {
                "effectivePermission": "NOT_PUBLIC",
                "permissionConfiguration": {
                    "accountLevelPermissions": {
                        "blockPublicAccess": {
                            "blockPublicAcls": true,
                            "blockPublicPolicy": true,
                            "ignorePublicAcls": true,
                            "restrictPublicBuckets": true
                        }
                    },
                    "bucketLevelPermissions": {
                        "accessControlList": {
```

```
                        "allowsPublicReadAccess": false,
                        "allowsPublicWriteAccess": false
                    },
                    "blockPublicAccess": {
                        "blockPublicAcls": true,
                        "blockPublicPolicy": true,
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    },
                    "bucketPolicy": {
                        "allowsPublicReadAccess": false,
                        "allowsPublicWriteAccess": false
                    }
                }
            }
        },
        "region": "us-west-2",
        "replicationDetails": {
            "replicated": false,
            "replicatedExternally": false,
            "replicationAccounts": []
        },
        "sensitivityScore": 95,
        "serverSideEncryption": {
            "kmsMasterKeyId": null,
            "type": "AES256"
        },
        "sharedAccess": "EXTERNAL",
        "sizeInBytes": 175978,
        "sizeInBytesCompressed": 0,
        "tags": [
            {
                "key": "Division",
                "value": "HR"
            },
            {
                "key": "Team",
                "value": "Recruiting"
            }
        ],
        "unclassifiableObjectCount": {
            "fileType": 3,
            "storageClass": 0,
            "total": 3
```

```
            },
            "unclassifiableObjectSizeInBytes": {
                "fileType": 2999826,
                "storageClass": 0,
                "total": 2999826
            },
            "versioning": true
        }
    ]
}
```

Pour plus d'informations, consultez Filtrer votre inventaire de compartiments S3 dans le guide de l'utilisateur Amazon Macie.

- Pour plus de détails sur l'API, voir DescribeBucketsla section Référence des AWS CLI commandes.

Exemples d'Amazon Managed Grafana utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Managed Grafana.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`list-workspaces`

L'exemple de code suivant montre comment utiliser`list-workspaces`.

AWS CLI

Pour répertorier les espaces de travail du compte dans la région spécifiée par les informations d'identification de l'utilisateur

L'`list-workspaces`exemple suivant répertorie les espaces de travail Grafana pour la région du compte.

```
aws grafana list-workspaces
```

Sortie :

```
{
    "workspaces": [
        {
            "authentication": {
                "providers": [
                    "AWS_SSO"
                ]
            },
            "created": "2022-04-04T16:20:21.796000-07:00",
            "description": "to test tags",
            "endpoint": "g-949e7b44df.grafana-workspace.us-east-1.amazonaws.com",
            "grafanaVersion": "8.2",
            "id": "g-949e7b44df",
            "modified": "2022-04-04T16:20:21.796000-07:00",
            "name": "testtag2",
            "notificationDestinations": [
                "SNS"
            ],
            "status": "ACTIVE"
        },
        {
            "authentication": {
                "providers": [
                    "AWS_SSO"
                ]
            },
            "created": "2022-04-20T10:22:15.115000-07:00",
            "description": "ww",
            "endpoint": "g-bffa51ed1b.grafana-workspace.us-east-1.amazonaws.com",
            "grafanaVersion": "8.2",
            "id": "g-bffa51ed1b",
```

```
            "modified": "2022-04-20T10:22:15.115000-07:00",
            "name": "ww",
            "notificationDestinations": [
                "SNS"
            ],
            "status": "ACTIVE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListWorkspacesà la section Référence des AWS CLI commandes.

MediaConnect exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with MediaConnect.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-flow-outputs`

L'exemple de code suivant montre comment utiliser`add-flow-outputs`.

AWS CLI

Pour ajouter des sorties à un flux

L'`add-flow-outputs`exemple suivant ajoute des sorties au flux spécifié.

```
aws mediaconnect add-flow-outputs \
```

```
--flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
--outputs Description='NYC
 stream',Destination=192.0.2.12,Name=NYC,Port=3333,Protocol=rtp-
fec,SmoothingLatency=100 Description='LA
 stream',Destination=203.0.113.9,Name=LA,Port=4444,Protocol=rtp-
fec,SmoothingLatency=100
```

Sortie :

```
{
    "Outputs": [
        {
            "Port": 3333,
            "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
            "Name": "NYC",
            "Description": "NYC stream",
            "Destination": "192.0.2.12",
            "Transport": {
                "Protocol": "rtp-fec",
                "SmoothingLatency": 100
            }
        },
        {
            "Port": 4444,
            "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
            "Name": "LA",
            "Description": "LA stream",
            "Destination": "203.0.113.9",
            "Transport": {
                "Protocol": "rtp-fec",
                "SmoothingLatency": 100
            }
        }
    ],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

Pour plus d'informations, consultez la section Ajouter des sorties à un flux dans le Guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous AddFlowOutputsà la section Référence des AWS CLI commandes.

`create-flow`

L'exemple de code suivant montre comment utiliser`create-flow`.

AWS CLI

Pour créer un flux

L'`create-flow`exemple suivant crée un flux avec la configuration spécifiée.

```
aws mediaconnect create-flow \
    --availability-zone us-west-2c \
    --name ExampleFlow \
    --source Description='Example source,
 backup',IngestPort=1055,Name=BackupSource,Protocol=rtp,WhitelistCidr=10.24.34.0/23
```

Sortie :

```
{
    "Flow": {
        "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:ExampleFlow",
        "AvailabilityZone": "us-west-2c",
        "EgressIp": "54.245.71.21",
        "Source": {
            "IngestPort": 1055,
            "SourceArn": "arn:aws:mediaconnect:us-
east-1:123456789012:source:2-3aBC45dEF67hiJ89-c34de5fG678h:BackupSource",
            "Transport": {
                "Protocol": "rtp",
                "MaxBitrate": 80000000
            },
            "Description": "Example source, backup",
            "IngestIp": "54.245.71.21",
            "WhitelistCidr": "10.24.34.0/23",
            "Name": "mySource"
        },
        "Entitlements": [],
        "Name": "ExampleFlow",
        "Outputs": [],
```

```
        "Status": "STANDBY",
        "Description": "Example source, backup"
    }
}
```

Pour plus d'informations, consultez la section Création d'un flux dans le guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateFlowà la section Référence des AWS CLI commandes.

`delete-flow`

L'exemple de code suivant montre comment utiliser`delete-flow`.

AWS CLI

Pour supprimer un flux

L'`delete-flow`exemple suivant supprime le flux spécifié.

```
aws mediaconnect delete-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Status": "DELETING"
}
```

Pour plus d'informations, consultez la section Suppression d'un flux dans le guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteFlowà la section Référence des AWS CLI commandes.

`describe-flow`

L'exemple de code suivant montre comment utiliser`describe-flow`.

AWS CLI

Pour afficher les détails d'un flux

L'`describe-flow`exemple suivant affiche les détails du flux spécifié, tels que l'ARN, la zone de disponibilité, le statut, la source, les droits et les sorties.

```
aws mediaconnect describe-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Sortie :

```
{
    "Flow": {
        "EgressIp": "54.201.4.39",
        "AvailabilityZone": "us-west-2c",
        "Status": "ACTIVE",
        "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
        "Entitlements": [
            {
                "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:123456789012:entitlement:1-AaBb11CcDd22EeFf-34DE5fG12AbC:MyEntitlement",
                "Description": "Assign to this account",
                "Name": "MyEntitlement",
                "Subscribers": [
                    "444455556666"
                ]
            }
        ],
        "Description": "NYC awards show",
        "Name": "AwardsShow",
        "Outputs": [
            {
                "Port": 2355,
                "Name": "NYC",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                },
                "OutputArn": "arn:aws:mediaconnect:us-
east-1:123456789012:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
```

```
                "Destination": "192.0.2.0"
            },
            {
                "Port": 3025,
                "Name": "LA",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                },
                "OutputArn": "arn:aws:mediaconnect:us-
east-1:123456789012:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
                "Destination": "192.0.2.0"
            }
        ],
        "Source": {
            "IngestIp": "54.201.4.39",
            "SourceArn": "arn:aws:mediaconnect:us-
east-1:123456789012:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
            "Transport": {
                "MaxBitrate": 80000000,
                "Protocol": "rtp"
            },
            "IngestPort": 1069,
            "Description": "Saturday night show",
            "Name": "ShowSource",
            "WhitelistCidr": "10.24.34.0/23"
        }
    }
}
```

Pour plus d'informations, consultez la section Affichage des détails d'un flux dans le guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribeFlowà la section Référence des AWS CLI commandes.

`grant-flow-entitlements`

L'exemple de code suivant montre comment utiliser`grant-flow-entitlements`.

AWS CLI

Pour accorder un droit à un flux

L'`grant-flow-entitlements`exemple suivant autorise le flux existant spécifié à partager votre contenu avec un autre AWS compte.

```
aws mediaconnect grant-flow-entitlements \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --entitlements Description='For
 AnyCompany',Encryption={"Algorithm=aes128,KeyType=static-
key,RoleArn=arn:aws:iam::111122223333:role/MediaConnect-
ASM,SecretArn=arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"},Name=AnyCompany_Entitlement,Subscribers=444455556666
 Description='For Example Corp',Name=ExampleCorp,Subscribers=777788889999
```

Sortie :

```
{
    "Entitlements": [
        {
            "Name": "AnyCompany_Entitlement",
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
            "Subscribers": [
                "444455556666"
            ],
            "Description": "For AnyCompany",
            "Encryption": {
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1",
                "Algorithm": "aes128",
                "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
                "KeyType": "static-key"
            }
        },
        {
            "Name": "ExampleCorp",
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-3333cccc4444dddd-1111aaaa2222:ExampleCorp",
            "Subscribers": [
                "777788889999"
            ],
            "Description": "For Example Corp"
        }
    ],
```

```
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

Pour plus d'informations, consultez la section Octroi d'un droit sur un flux dans le guide de l' MediaConnect utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous GrantFlowEntitlementsà la section Référence des AWS CLI commandes.

`list-entitlements`

L'exemple de code suivant montre comment utiliser`list-entitlements`.

AWS CLI

Pour consulter la liste des droits

L'`list-entitlements`exemple suivant affiche une liste de tous les droits accordés au compte.

```
aws mediaconnect list-entitlements
```

Sortie :

```
{
    "Entitlements": [
        {
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
            "EntitlementName": "MyEntitlement"
        }
    ]
}
```

Pour plus d'informations, consultez ListEntitlementsle manuel AWS Elemental MediaConnect API Reference.

- Pour plus de détails sur l'API, reportez-vous ListEntitlementsà la section Référence des AWS CLI commandes.

`list-flows`

L'exemple de code suivant montre comment utiliser`list-flows`.

AWS CLI

Pour consulter la liste des flux

L'`list-flows`exemple suivant affiche une liste de flux.

```
aws mediaconnect list-flows
```

Sortie :

```
{
    "Flows": [
        {
            "Status": "STANDBY",
            "SourceType": "OWNED",
            "AvailabilityZone": "us-west-2a",
            "Description": "NYC awards show",
            "Name": "AwardsShow",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
        },
        {
            "Status": "STANDBY",
            "SourceType": "OWNED",
            "AvailabilityZone": "us-west-2c",
            "Description": "LA basketball game",
            "Name": "BasketballGame",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste de flux dans le guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListFlowsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une MediaConnect ressource

L'`list-tags-for-resource`exemple suivant affiche les clés de balise et les valeurs associées à la MediaConnect ressource spécifiée.

```
aws mediaconnect list-tags-for-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
```

Sortie :

```
{
    "Tags": {
        "region": "west",
        "stage": "prod"
    }
}
```

Pour plus d'informations, voir ListTagsForResource TagResource, UntagResource dans le manuel AWS Elemental MediaConnect API Reference.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`remove-flow-output`

L'exemple de code suivant montre comment utiliser`remove-flow-output`.

AWS CLI

Pour supprimer une sortie d'un flux

L'`remove-flow-output`exemple suivant supprime une sortie du flux spécifié.

```
aws mediaconnect remove-flow-output \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
```

```
    --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC
```

Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC"
}
```

Pour plus d'informations, consultez la section Suppression des sorties d'un flux dans le Guide de MediaConnect l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous RemoveFlowOutputà la section Référence des AWS CLI commandes.

`revoke-flow-entitlement`

L'exemple de code suivant montre comment utiliser`revoke-flow-entitlement`.

AWS CLI

Pour révoquer un droit

L'`revoke-flow-entitlement`exemple suivant révoque un droit sur le flux spécifié.

```
aws mediaconnect revoke-flow-entitlement \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
```

Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement"
}
```

Pour plus d'informations, consultez la section Révocation d'un droit dans le guide de l'utilisateur AWS Elemental MediaConnect .

- Pour plus de détails sur l'API, reportez-vous RevokeFlowEntitlementà la section Référence des AWS CLI commandes.

`start-flow`

L'exemple de code suivant montre comment utiliser`start-flow`.

AWS CLI

Pour démarrer un flux

L'`start-flow`exemple suivant démarre le flux spécifié.

```
aws mediaconnect start-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Status": "STARTING"
}
```

Pour plus d'informations, consultez la section Démarrage d'un flux dans le guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous StartFlowà la section Référence des AWS CLI commandes.

`stop-flow`

L'exemple de code suivant montre comment utiliser`stop-flow`.

AWS CLI

Pour arrêter un flux

L'`stop-flow`exemple suivant arrête le flux spécifié.

```
aws mediaconnect stop-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Sortie :

```
{
    "Status": "STOPPING",
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
}
```

Pour plus d'informations, consultez la section Stopping a Flow dans le Guide de MediaConnect l'utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous StopFlowà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une MediaConnect ressource

L'`tag-resource`exemple suivant ajoute une balise avec un nom et une valeur de clé à la MediaConnect ressource spécifiée.

```
aws mediaconnect tag-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
    --tags region=west
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir ListTagsForResource TagResource, UntagResource dans le manuel AWS Elemental MediaConnect API Reference.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une MediaConnect ressource

L'`untag-resource`exemple suivant supprime la balise portant le nom de clé spécifié et la valeur associée d'une MediaConnect ressource.

```
aws mediaconnect untag-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \
    --tag-keys region
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir ListTagsForResource TagResource, UntagResource dans le manuel AWS Elemental MediaConnect API Reference.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-flow-entitlement`

L'exemple de code suivant montre comment utiliser`update-flow-entitlement`.

AWS CLI

Pour mettre à jour un droit

L'`update-flow-entitlement`exemple suivant met à jour le droit spécifié avec une nouvelle description et un nouvel abonné.

```
aws mediaconnect update-flow-entitlement \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
```

```
    --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
 \
    --description 'For AnyCompany Affiliate' \
    --subscribers 777788889999
```

Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "Entitlement": {
        "Name": "AnyCompany_Entitlement",
        "Description": "For AnyCompany Affiliate",
        "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
        "Encryption": {
            "KeyType": "static-key",
            "Algorithm": "aes128",
            "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
            "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
        },
        "Subscribers": [
            "777788889999"
        ]
    }
}
```

Pour plus d'informations, consultez la section Mise à jour d'un droit dans le guide de l' MediaConnect utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous UpdateFlowEntitlementà la section Référence des AWS CLI commandes.

`update-flow-output`

L'exemple de code suivant montre comment utiliser`update-flow-output`.

AWS CLI

Pour mettre à jour une sortie sur un flux

L'`update-flow-output`exemple suivant met à jour une sortie sur le flux spécifié.

```
aws mediaconnect update-flow-output \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC \
    --port 3331
```

Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "Output": {
        "Name": "NYC",
        "Port": 3331,
        "Description": "NYC stream",
        "Transport": {
            "Protocol": "rtp-fec",
            "SmoothingLatency": 100
        },
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
        "Destination": "192.0.2.12"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour des sorties d'un flux dans le Guide de MediaConnect l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous UpdateFlowOutputà la section Référence des AWS CLI commandes.

`update-flow-source`

L'exemple de code suivant montre comment utiliser`update-flow-source`.

AWS CLI

Pour mettre à jour la source d'un flux existant

L'`update-flow-source`exemple suivant met à jour la source d'un flux existant.

```
aws mediaconnect update-flow-source \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \
    --source-arn arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource \
    --description 'Friday night show' \
    --ingest-port 3344 \
    --protocol rtp-fec \
    --whitelist-cidr 10.24.34.0/23
```

Sortie :

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Source": {
        "IngestIp": "34.210.136.56",
        "WhitelistCidr": "10.24.34.0/23",
        "Transport": {
            "Protocol": "rtp-fec"
        },
        "IngestPort": 3344,
        "Name": "ShowSource",
        "Description": "Friday night show",
        "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"
    }
}
```

Pour plus d'informations, consultez la section Mise à jour de la source d'un flux dans le Guide de MediaConnect l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous UpdateFlowSourceà la section Référence des AWS CLI commandes.

MediaConvert exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with MediaConvert.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-job`

L'exemple de code suivant montre comment utiliser`cancel-job`.

AWS CLI

Pour annuler une tâche figurant dans une file d'attente

L'`cancel-job`exemple suivant annule la tâche avec ID. `1234567891234-abc123` Vous ne pouvez pas annuler une tâche que le service a commencé à traiter.

```
aws mediaconvert cancel-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --id 1234567891234-abc123
```

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Pour plus d'informations, consultez la section Travailler avec des MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir CancelJobla section Référence des AWS CLI commandes.

`create-job-template`

L'exemple de code suivant montre comment utiliser`create-job-template`.

AWS CLI

Création d'un modèle de tâche

L'`create-job-template`exemple suivant crée un modèle de tâche avec les paramètres de transcodage spécifiés dans le fichier `job-template.json` qui réside sur votre système.

```
aws mediaconvert create-job-template \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --name JobTemplate1 \
    --cli-input-json file://~/job-template.json
```

Si vous créez le fichier JSON de modèle de tâche en utilisant `get-job-template` puis en modifiant le fichier, supprimez l'`JobTemplate`objet, mais conservez l'objet enfant Settings à l'intérieur. Assurez-vous également de supprimer les paires clé-valeur suivantes :`LastUpdated`, `ArnType`, et. `CreatedAt` Vous pouvez spécifier la catégorie, la description, le nom et la file d'attente dans le fichier JSON ou sur la ligne de commande.

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Si votre demande aboutit, le service renvoie la spécification JSON pour le modèle de tâche que vous avez créé.

Pour plus d'informations, consultez la section Travailler avec des modèles de MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir CreateJobTemplatela section Référence des AWS CLI commandes.

`create-job`

L'exemple de code suivant montre comment utiliser`create-job`.

AWS CLI

Pour créer un emploi

L'`create-job`exemple suivant crée une tâche de transcodage avec les paramètres spécifiés dans un fichier `job.json` résidant sur le système à partir duquel vous envoyez la commande.

Cette spécification de tâche JSON peut spécifier chaque paramètre individuellement, faire référence à un modèle de tâche ou à des préréglages de sortie.

```
aws mediaconvert create-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --cli-input-json file://~/job.json
```

Vous pouvez utiliser la MediaConvert console AWS Elemental pour générer la spécification de tâche JSON en choisissant les paramètres de votre tâche, puis en choisissant Afficher le JSON de la tâche en bas de la section Job.

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Si votre demande aboutit, le service renvoie la spécification de tâche JSON que vous avez envoyée avec votre demande.

Pour plus d'informations, consultez la section Travailler avec des MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir CreateJobla section Référence des AWS CLI commandes.

`create-preset`

L'exemple de code suivant montre comment utiliser`create-preset`.

AWS CLI

Pour créer un préréglage de sortie personnalisé

L'`create-preset`exemple suivant crée un préréglage de sortie personnalisé basé sur les paramètres de sortie spécifiés dans le fichier`preset.json`. Vous pouvez spécifier la catégorie, la description et le nom dans le fichier JSON ou sur la ligne de commande.

```
aws mediaconvert create-preset \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com
    --region region-name-1 \
    --cli-input-json file://~/preset.json
```

Si vous créez votre fichier JSON prédéfini en utilisant `get-preset` puis en modifiant le fichier de sortie, assurez-vous de supprimer les paires clé-valeur suivantes :`LastUpdated`, `ArnType`, et. `CreatedAt`

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Pour plus d'informations, consultez la section Utilisation des préréglages de MediaConvert sortie AWS élémentaires dans le guide de l'utilisateur d'AWS Elemental MediaConvert .

- Pour plus de détails sur l'API, voir CreatePresetla section Référence des AWS CLI commandes.

`create-queue`

L'exemple de code suivant montre comment utiliser`create-queue`.

AWS CLI

Pour créer une file d'attente personnalisée

L'`create-queue`exemple suivant crée une file d'attente de transcodage personnalisée.

```
aws mediaconvert create-queue \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --name Queue1 \
    --description "Keep this queue empty unless job is urgent."
```

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Sortie :

```
{
    "Queue": {
        "Status": "ACTIVE",
        "Name": "Queue1",
        "LastUpdated": 1518034928,
        "Arn": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",
        "Type": "CUSTOM",
```

```
        "CreatedAt": 1518034928,
        "Description": "Keep this queue empty unless job is urgent."
    }
}
```

Pour plus d'informations, consultez la section Utilisation des MediaConvert files d'attente AWS élémentaires dans le guide de l'utilisateur d'AWSElemental MediaConvert .

- Pour plus de détails sur l'API, voir CreateQueuela section Référence des AWS CLI commandes.

`delete-job-template`

L'exemple de code suivant montre comment utiliser`delete-job-template`.

AWS CLI

Pour supprimer un modèle de tâche

L'`delete-job-template`exemple suivant supprime le modèle de tâche personnalisé spécifié.

```
aws mediaconvert delete-job-template \
    --name "DASH Streaming" \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Cette commande ne produit aucun résultat. Exécutez `aws mediaconvert list-job-templates` pour confirmer que votre modèle a été supprimé.

Pour plus d'informations, consultez la section Travailler avec des modèles de MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir DeleteJobTemplatela section Référence des AWS CLI commandes.

`delete-preset`

L'exemple de code suivant montre comment utiliser`delete-preset`.

AWS CLI

Pour supprimer une file d'attente à la demande personnalisée

L'`delete-preset`exemple suivant supprime le préréglage personnalisé spécifié.

```
aws mediaconvert delete-preset \
    --name SimpleMP4 \
    --endpoint-url  https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Cette commande ne produit aucun résultat. Exécutez `aws mediaconvert list-presets` pour confirmer que votre préréglage a été supprimé.

Pour plus d'informations, consultez la section Utilisation des préréglages de MediaConvert sortie AWS élémentaires dans le guide de l'utilisateur d'AWS Elemental MediaConvert .

- Pour plus de détails sur l'API, voir DeletePresetla section Référence des AWS CLI commandes.

`delete-queue`

L'exemple de code suivant montre comment utiliser`delete-queue`.

AWS CLI

Pour supprimer une file d'attente à la demande personnalisée

L'`delete-queue`exemple suivant supprime la file d'attente à la demande personnalisée spécifiée.

Vous ne pouvez pas supprimer votre file d'attente par défaut. Vous ne pouvez pas supprimer une file d'attente réservée qui possède un plan de tarification active ou qui contient des tâches non traitées.

```
aws mediaconvert delete-queue \
    --name Customer1 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Cette commande ne produit aucun résultat. Exécutez `aws mediaconvert list-queues` pour confirmer que votre file d'attente a été supprimée.

Pour plus d'informations, consultez la section Utilisation des MediaConvert files d'attente AWS élémentaires dans le guide de l'utilisateur d'AWSElemental MediaConvert .

- Pour plus de détails sur l'API, voir DeleteQueuela section Référence des AWS CLI commandes.

`describe-endpoints`

L'exemple de code suivant montre comment utiliser`describe-endpoints`.

AWS CLI

Pour obtenir un point de terminaison spécifique à votre compte

L'`describe-endpoints`exemple suivant récupère le point de terminaison dont vous avez besoin pour envoyer toute autre demande au service.

```
aws mediaconvert describe-endpoints
```

Sortie :

```
{
    "Endpoints": [
        {
            "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, voir Commencer à MediaConvert utiliser l'API dans le manuel AWS Elemental MediaConvert API Reference.

- Pour plus de détails sur l'API, voir DescribeEndpointsla section Référence des AWS CLI commandes.

`get-job-template`

L'exemple de code suivant montre comment utiliser`get-job-template`.

AWS CLI

Pour obtenir les détails d'un modèle de poste

L'`get-job-template`exemple suivant affiche la définition JSON du modèle de tâche personnalisé spécifié.

```
aws mediaconvert get-job-template \
    --name "DASH Streaming" \
    --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```

Sortie :

```
{
    "JobTemplate": {
        "StatusUpdateInterval": "SECONDS_60",
        "LastUpdated": 1568652998,
        "Description": "Create a DASH streaming ABR stack",
        "CreatedAt": 1568652998,
        "Priority": 0,
        "Name": "DASH Streaming",
        "Settings": {
            ...<truncatedforbrevity>...
        },
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH
 Streaming",
        "Type": "CUSTOM"
    }
}
```

Pour plus d'informations, consultez la section Travailler avec des modèles de MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir GetJobTemplatela section Référence des AWS CLI commandes.

`get-job`

L'exemple de code suivant montre comment utiliser`get-job`.

AWS CLI

Pour obtenir des informations sur un poste en particulier

L'exemple suivant demande les informations relatives à la tâche avec ID`1234567890987-1ab2c3`, ce qui, dans cet exemple, s'est soldé par une erreur.

```
aws mediaconvert get-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --id 1234567890987-1ab2c3
```

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Si votre demande aboutit, le service renvoie un fichier JSON contenant les informations relatives à la tâche, notamment les paramètres de la tâche, les erreurs renvoyées et d'autres données relatives à la tâche, comme suit :

```
{
    "Job": {
        "Status": "ERROR",
        "Queue": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",
        "Settings": {
            ...<truncated for brevity>...
        },
        "ErrorMessage": "Unable to open input file [s3://my-input-bucket/file-
name.mp4]: [Failed probe/open: [Failed to read data: AssumeRole failed]]",
        "ErrorCode": 1434,
        "Role": "arn:aws:iam::012345678998:role/MediaConvertServiceRole",
        "Arn": "arn:aws:mediaconvert:us-
west-1:012345678998:jobs/1234567890987-1ab2c3",
        "UserMetadata": {},
        "Timing": {
            "FinishTime": 1517442131,
            "SubmitTime": 1517442103,
            "StartTime": 1517442104
        },
        "Id": "1234567890987-1ab2c3",
        "CreatedAt": 1517442103
    }
}
```

Pour plus d'informations, consultez la section Travailler avec des MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir GetJobla section Référence des AWS CLI commandes.

`get-preset`

L'exemple de code suivant montre comment utiliser`get-preset`.

AWS CLI

Pour obtenir des informations sur un préréglage en particulier

L'`get-preset`exemple suivant demande la définition JSON du préréglage personnalisé spécifié.

```
aws mediaconvert get-preset \
    --name SimpleMP4 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Preset": {
        "Description": "Creates basic MP4 file. No filtering or preproccessing.",
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4",
        "LastUpdated": 1568843141,
        "Name": "SimpleMP4",
        "Settings": {
            "ContainerSettings": {
                "Mp4Settings": {
                    "FreeSpaceBox": "EXCLUDE",
                    "CslgAtom": "INCLUDE",
                    "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
                },
                "Container": "MP4"
            },
            "AudioDescriptions": [
                {
                    "LanguageCodeControl": "FOLLOW_INPUT",
                    "AudioTypeControl": "FOLLOW_INPUT",
                    "CodecSettings": {
                        "AacSettings": {
                            "RawFormat": "NONE",
                            "CodecProfile": "LC",
                            "AudioDescriptionBroadcasterMix": "NORMAL",
                            "SampleRate": 48000,
                            "Bitrate": 96000,
                            "RateControlMode": "CBR",
                            "Specification": "MPEG4",
                            "CodingMode": "CODING_MODE_2_0"
                        },
                        "Codec": "AAC"
                    }
                }
            ],
            "VideoDescription": {
                "RespondToAfd": "NONE",
                "TimecodeInsertion": "DISABLED",
```

```
                "Sharpness": 50,
                "ColorMetadata": "INSERT",
                "CodecSettings": {
                    "H264Settings": {
                        "FramerateControl": "INITIALIZE_FROM_SOURCE",
                        "SpatialAdaptiveQuantization": "ENABLED",
                        "Softness": 0,
                        "Telecine": "NONE",
                        "CodecLevel": "AUTO",
                        "QualityTuningLevel": "SINGLE_PASS",
                        "UnregisteredSeiTimecode": "DISABLED",
                        "Slices": 1,
                        "Syntax": "DEFAULT",
                        "GopClosedCadence": 1,
                        "AdaptiveQuantization": "HIGH",
                        "EntropyEncoding": "CABAC",
                        "InterlaceMode": "PROGRESSIVE",
                        "ParControl": "INITIALIZE_FROM_SOURCE",
                        "NumberBFramesBetweenReferenceFrames": 2,
                        "GopSizeUnits": "FRAMES",
                        "RepeatPps": "DISABLED",
                        "CodecProfile": "MAIN",
                        "FieldEncoding": "PAFF",
                        "GopSize": 90.0,
                        "SlowPal": "DISABLED",
                        "SceneChangeDetect": "ENABLED",
                        "GopBReference": "DISABLED",
                        "RateControlMode": "CBR",
                        "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                        "FlickerAdaptiveQuantization": "DISABLED",
                        "DynamicSubGop": "STATIC",
                        "MinIInterval": 0,
                        "TemporalAdaptiveQuantization": "ENABLED",
                        "Bitrate": 400000,
                        "NumberReferenceFrames": 3
                    },
                    "Codec": "H_264"
                },
                "AfdSignaling": "NONE",
                "AntiAlias": "ENABLED",
                "ScalingBehavior": "DEFAULT",
                "DropFrameTimecode": "ENABLED"
            }
        },
```

```
        "Type": "CUSTOM",
        "CreatedAt": 1568841521
    }
}
```

Pour plus d'informations, consultez la section Utilisation des préréglages de MediaConvert sortie AWS élémentaires dans le guide de l'utilisateur d'AWS Elemental MediaConvert .

- Pour plus de détails sur l'API, voir GetPresetla section Référence des AWS CLI commandes.

`get-queue`

L'exemple de code suivant montre comment utiliser`get-queue`.

AWS CLI

Pour obtenir les détails d'une file d'attente

L'`get-queue`exemple suivant récupère les détails de la file d'attente personnalisée spécifiée.

```
aws mediaconvert get-queue \
    --name Customer1 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Queue": {
        "LastUpdated": 1526428502,
        "Type": "CUSTOM",
        "SubmittedJobsCount": 0,
        "Status": "ACTIVE",
        "PricingPlan": "ON_DEMAND",
        "CreatedAt": 1526428502,
        "ProgressingJobsCount": 0,
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
        "Name": "Customer1"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des MediaConvert files d'attente AWS élémentaires dans le guide de l'utilisateur d'AWSElemental MediaConvert .

- Pour plus de détails sur l'API, voir GetQueuela section Référence des AWS CLI commandes.

`list-job-templates`

L'exemple de code suivant montre comment utiliser`list-job-templates`.

AWS CLI

Exemple 1 : pour répertorier vos modèles de tâches personnalisés

L'`list-job-templates`exemple suivant répertorie tous les modèles de tâches personnalisés de la région actuelle. Pour répertorier les modèles de tâches du système, reportez-vous à l'exemple suivant.

```
aws mediaconvert list-job-templates \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "JobTemplates": [
        {
            "Description": "Create a DASH streaming ABR stack",
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH
 Streaming",
            "Name": "DASH Streaming",
            "LastUpdated": 1568653007,
            "Priority": 0,
            "Settings": {
                ...<truncatedforbrevity>...
            },
            "Type": "CUSTOM",
            "StatusUpdateInterval": "SECONDS_60",
            "CreatedAt": 1568653007
        },
        {
            "Description": "Create a high-res file",
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/File",
            "Name": "File",
            "LastUpdated": 1568653007,
            "Priority": 0,
            "Settings": {
```

```
                ...<truncatedforbrevity>...
            },
            "Type": "CUSTOM",
            "StatusUpdateInterval": "SECONDS_60",
            "CreatedAt": 1568653023
        }
    ]
}
```

Exemple 2 : pour répertorier les modèles de tâches MediaConvert du système

L'`list-job-templates`exemple suivant répertorie tous les modèles de tâches du système.

```
aws mediaconvert list-job-templates \
    --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \
    --list-by SYSTEM
```

Sortie :

```
{
    "JobTemplates": [
        {
            "CreatedAt": 1568321779,
            "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:jobTemplates/System-
Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
            "Name": "System-Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
            "Description": "GENERIC, MP4, AVC + HEV1(HEVC,SDR), AAC, SDR, QVBR",
            "Category": "GENERIC",
            "Settings": {
                "AdAvailOffset": 0,
                "OutputGroups": [
                    {
                        "Outputs": [
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
```

```
                                "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_0.8Mbps_Qvbr_Vq7",
                                "NameModifier":
 "_Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_800Kbps_Qvbr_Vq7"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12000Kbps_Qvbr_Vq9"
                            }
                        ],
                        "OutputGroupSettings": {
                            "FileGroupSettings": {

                            },
                            "Type": "FILE_GROUP_SETTINGS"
                        },
                        "Name": "File Group"
                    }
                ]
            },
```

```
            "Type": "SYSTEM",
            "LastUpdated": 1568321779
        },
        ...<truncatedforbrevity>...
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec des modèles de MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir ListJobTemplatesla section Référence des AWS CLI commandes.

`list-jobs`

L'exemple de code suivant montre comment utiliser`list-jobs`.

AWS CLI

Pour obtenir des informations sur tous les emplois d'une région

L'exemple suivant demande les informations relatives à toutes vos tâches dans la région spécifiée.

```
aws mediaconvert list-jobs \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1
```

Pour obtenir un point de terminaison spécifique à votre compte`describe-endpoints`, utilisez ou envoyez la commande sans le point de terminaison. Le service renvoie une erreur et renvoie votre point de terminaison.

Pour plus d'informations, consultez la section Travailler avec des MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir ListJobsla section Référence des AWS CLI commandes.

`list-presets`

L'exemple de code suivant montre comment utiliser`list-presets`.

AWS CLI

Exemple 1 : pour répertorier vos préréglages de sortie personnalisés

L'`list-presets`exemple suivant répertorie vos préréglages de sortie personnalisés. Pour répertorier les préréglages du système, reportez-vous à l'exemple suivant.

```
aws mediaconvert list-presets \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Presets": [
        {
            "Name": "SimpleMP4",
            "CreatedAt": 1568841521,
            "Settings": {
                ......
            },
            "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",
            "Type": "CUSTOM",
            "LastUpdated": 1568843141,
            "Description": "Creates basic MP4 file. No filtering or preproccessing."
        },
        {
            "Name": "SimpleTS",
            "CreatedAt": 1568843113,
            "Settings": {
                ... truncated for brevity ...
            },
            "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleTS",
            "Type": "CUSTOM",
            "LastUpdated": 1568843113,
            "Description": "Create a basic transport stream."
        }
    ]
}
```

Exemple 2 : pour répertorier les préréglages de sortie du système

L'`list-presets`exemple suivant répertorie les préréglages MediaConvert système disponibles. Pour répertorier vos préréglages personnalisés, consultez l'exemple précédent.

```
aws mediaconvert list-presets \
    --list-by SYSTEM \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Presets": [
        {
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/System-
Avc_16x9_1080p_29_97fps_8500kbps",
            "Name": "System-Avc_16x9_1080p_29_97fps_8500kbps",
            "CreatedAt": 1568321789,
            "Description": "Wifi, 1920x1080, 16:9, 29.97fps, 8500kbps",
            "LastUpdated": 1568321789,
            "Type": "SYSTEM",
            "Category": "HLS",
            "Settings": {
            ...<output settings removed for brevity>...
            }
        },

        ...<list of presets shortened for brevity>...

        {
            "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:presets/System-
Xdcam_HD_1080i_29_97fps_35mpbs",
            "Name": "System-Xdcam_HD_1080i_29_97fps_35mpbs",
            "CreatedAt": 1568321790,
            "Description": "XDCAM MPEG HD, 1920x1080i, 29.97fps, 35mbps",
            "LastUpdated": 1568321790,
            "Type": "SYSTEM",
            "Category": "MXF",
            "Settings": {
            ...<output settings removed for brevity>...
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des préréglages de MediaConvert sortie AWS élémentaires dans le guide de l'utilisateur d'AWS Elemental MediaConvert .

- Pour plus de détails sur l'API, voir ListPresetsla section Référence des AWS CLI commandes.

`list-queues`

L'exemple de code suivant montre comment utiliser`list-queues`.

AWS CLI

Pour répertorier vos files d'attente

L'`list-queues`exemple suivant répertorie toutes vos MediaConvert files d'attente.

```
aws mediaconvert list-queues \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Queues": [
        {
            "PricingPlan": "ON_DEMAND",
            "Type": "SYSTEM",
            "Status": "ACTIVE",
            "CreatedAt": 1503451595,
            "Name": "Default",
            "SubmittedJobsCount": 0,
            "ProgressingJobsCount": 0,
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Default",
            "LastUpdated": 1534549158
        },
        {
            "PricingPlan": "ON_DEMAND",
            "Type": "CUSTOM",
            "Status": "ACTIVE",
            "CreatedAt": 1537460025,
            "Name": "Customer1",
            "SubmittedJobsCount": 0,
            "Description": "Jobs we run for our cusotmer.",
            "ProgressingJobsCount": 0,
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
            "LastUpdated": 1537460025
        },
        {
```

```
            "ProgressingJobsCount": 0,
            "Status": "ACTIVE",
            "Name": "transcode-library",
            "SubmittedJobsCount": 0,
            "LastUpdated": 1564066204,
            "ReservationPlan": {
                "Status": "ACTIVE",
                "ReservedSlots": 1,
                "PurchasedAt": 1564066203,
                "Commitment": "ONE_YEAR",
                "ExpiresAt": 1595688603,
                "RenewalType": "EXPIRE"
            },
            "PricingPlan": "RESERVED",
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/transcode-
library",
            "Type": "CUSTOM",
            "CreatedAt": 1564066204
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des MediaConvert files d'attente AWS élémentaires dans le guide de l'utilisateur d'AWSElemental MediaConvert .

- Pour plus de détails sur l'API, voir ListQueuesla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises MediaConvert d'une file d'attente, d'un modèle de tâche ou d'un préréglage de sortie

L'`list-tags-for-resource`exemple suivant répertorie les balises du préréglage de sortie spécifié.

```
aws mediaconvert list-tags-for-resource \
    --arn arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "ResourceTags": {
        "Tags": {
            "customer": "zippyVideo"
        },
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"
    }
}
```

Pour plus d'informations, consultez la section Marquage des files d' MediaConvert attente AWS élémentaires, des modèles de tâches et des préréglages de sortie dans le guide de l'AWS utilisateur d'Elemental. MediaConvert

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`update-job-template`

L'exemple de code suivant montre comment utiliser`update-job-template`.

AWS CLI

Pour modifier un modèle de tâche

L'`update-job-template`exemple suivant remplace la définition JSON du modèle de tâche personnalisé spécifié par la définition JSON du fichier fourni.

aws mediaconvert update-job-template --name File1 https://abcd1234.mediaconvert.us-west-2.amazonaws.com --endpoint-url -- fichier : //~/ .json cli-input-json job-template-update

Contenu de `job-template-update.json` :

```
{
    "Description": "A simple job template that generates a single file output.",
    "Queue": "arn:aws:mediaconvert:us-east-1:012345678998:queues/Default",
    "Name": "SimpleFile",
    "Settings": {
      "OutputGroups": [
        {
          "Name": "File Group",
          "Outputs": [
```

```
            {
              "ContainerSettings": {
                "Container": "MP4",
                "Mp4Settings": {
                  "CslgAtom": "INCLUDE",
                  "FreeSpaceBox": "EXCLUDE",
                  "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
                }
              },
              "VideoDescription": {
                "ScalingBehavior": "DEFAULT",
                "TimecodeInsertion": "DISABLED",
                "AntiAlias": "ENABLED",
                "Sharpness": 50,
                "CodecSettings": {
                  "Codec": "H_264",
                  "H264Settings": {
                    "InterlaceMode": "PROGRESSIVE",
                    "NumberReferenceFrames": 3,
                    "Syntax": "DEFAULT",
                    "Softness": 0,
                    "GopClosedCadence": 1,
                    "GopSize": 90,
                    "Slices": 1,
                    "GopBReference": "DISABLED",
                    "SlowPal": "DISABLED",
                    "SpatialAdaptiveQuantization": "ENABLED",
                    "TemporalAdaptiveQuantization": "ENABLED",
                    "FlickerAdaptiveQuantization": "DISABLED",
                    "EntropyEncoding": "CABAC",
                    "Bitrate": 400000,
                    "FramerateControl": "INITIALIZE_FROM_SOURCE",
                    "RateControlMode": "CBR",
                    "CodecProfile": "MAIN",
                    "Telecine": "NONE",
                    "MinIInterval": 0,
                    "AdaptiveQuantization": "HIGH",
                    "CodecLevel": "AUTO",
                    "FieldEncoding": "PAFF",
                    "SceneChangeDetect": "ENABLED",
                    "QualityTuningLevel": "SINGLE_PASS",
                    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                    "UnregisteredSeiTimecode": "DISABLED",
                    "GopSizeUnits": "FRAMES",
```

```
                    "ParControl": "INITIALIZE_FROM_SOURCE",
                    "NumberBFramesBetweenReferenceFrames": 2,
                    "RepeatPps": "DISABLED",
                    "DynamicSubGop": "STATIC"
                  }
                },
                "AfdSignaling": "NONE",
                "DropFrameTimecode": "ENABLED",
                "RespondToAfd": "NONE",
                "ColorMetadata": "INSERT"
              },
              "AudioDescriptions": [
                {
                  "AudioTypeControl": "FOLLOW_INPUT",
                  "CodecSettings": {
                    "Codec": "AAC",
                    "AacSettings": {
                      "AudioDescriptionBroadcasterMix": "NORMAL",
                      "Bitrate": 96000,
                      "RateControlMode": "CBR",
                      "CodecProfile": "LC",
                      "CodingMode": "CODING_MODE_2_0",
                      "RawFormat": "NONE",
                      "SampleRate": 48000,
                      "Specification": "MPEG4"
                    }
                  },
                  "LanguageCodeControl": "FOLLOW_INPUT"
                }
              ]
            }
          ],
          "OutputGroupSettings": {
            "Type": "FILE_GROUP_SETTINGS",
            "FileGroupSettings": {}
          }
        }
      ],
      "AdAvailOffset": 0
    },
    "StatusUpdateInterval": "SECONDS_60",
    "Priority": 0
}
```

Le système renvoie la charge utile JSON que vous envoyez avec votre demande, même si celle-ci entraîne une erreur. Par conséquent, le JSON renvoyé n'est pas nécessairement la nouvelle définition du modèle de tâche.

La charge utile JSON pouvant être longue, vous devrez peut-être faire défiler la page vers le haut pour voir les messages d'erreur éventuels.

Pour plus d'informations, consultez la section Travailler avec des modèles de MediaConvert tâches AWS élémentaires dans le guide de l' MediaConvert utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, voir UpdateJobTemplatela section Référence des AWS CLI commandes.

`update-preset`

L'exemple de code suivant montre comment utiliser`update-preset`.

AWS CLI

Pour modifier un préréglage

L'`update-preset`exemple suivant remplace la description du préréglage spécifié.

```
aws mediaconvert update-preset \
--name Customer1 \
--description "New description text."
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "Preset": {
        "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",
        "Settings": {
        ...<output settings removed for brevity>...
        },
        "Type": "CUSTOM",
        "LastUpdated": 1568938411,
        "Description": "New description text.",
        "Name": "SimpleMP4",
        "CreatedAt": 1568938240
    }
```

```
}
```

Pour plus d'informations, consultez la section Utilisation des préréglages de MediaConvert sortie AWS élémentaires dans le guide de l'utilisateur d'AWS Elemental MediaConvert .

- Pour plus de détails sur l'API, voir UpdatePresetla section Référence des AWS CLI commandes.

`update-queue`

L'exemple de code suivant montre comment utiliser`update-queue`.

AWS CLI

Pour modifier une file d'attente

L'`update-queue`exemple suivant met en pause la file d'attente spécifiée en modifiant son statut en`PAUSED`.

```
aws mediaconvert update-queue \
--name Customer1 \
--status PAUSED
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Queue": {
        "LastUpdated": 1568839845,
        "Status": "PAUSED",
        "ProgressingJobsCount": 0,
        "CreatedAt": 1526428516,
        "Arn": "arn:aws:mediaconvert:us-west-1:123456789012:queues/Customer1",
        "Name": "Customer1",
        "SubmittedJobsCount": 0,
        "PricingPlan": "ON_DEMAND",
        "Type": "CUSTOM"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des MediaConvert files d'attente AWS élémentaires dans le guide de l'utilisateur d'AWSElemental MediaConvert .

- Pour plus de détails sur l'API, voir UpdateQueuela section Référence des AWS CLI commandes.

MediaLive exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with MediaLive.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-channel`

L'exemple de code suivant montre comment utiliser`create-channel`.

AWS CLI

Pour créer une chaîne

L'`create-channel`exemple suivant crée un canal en transmettant un fichier JSON contenant les paramètres que vous souhaitez spécifier.

Dans cet exemple, le canal ingère une entrée HLS PULL qui se connecte à une source contenant de la vidéo, du son et des sous-titres intégrés. Le canal crée un groupe de sortie HLS avec un serveur Akamai comme destination. Le groupe de sorties contient deux sorties : une pour la vidéo H.265 et l'audio AAC, et une pour les sous-titres Web-VTT, en anglais uniquement.

Le JSON de cet exemple de canal inclut les paramètres minimaux requis pour un canal qui utilise une entrée HLS PULL et qui produit un groupe de sortie HLS avec Akamai comme destination. Le JSON contient les sections principales suivantes :

`InputAttachments`, qui indique une source pour le son et une source pour les sous-titres. Il ne spécifie pas de sélecteur vidéo, ce qui signifie qu'il MediaLive extrait la première vidéo trouvée dans la source. `Destinations`, qui contient les deux adresses IP (URLs) pour le groupe de sortie unique de ce canal. Ces adresses nécessitent un mot de passe. `EncoderSettings`, qui contient des sous-sections. `AudioDescriptions`, qui indique que le canal contient une ressource de sortie audio, qui utilise la source de InputAttachments et produit du son au format AAC. `CaptionDescriptions`, qui indique que le canal contient une ressource de sortie de sous-titres, qui utilise la source de InputAttachments et produit des sous-titres au format Web-VTT. `VideoDescriptions`, qui indique que le canal contient une ressource de sortie vidéo, avec la résolution spécifiée. `OutputGroups`, qui spécifie les groupes de sortie. Dans cet exemple, un groupe est nomméAkamai. La connexion est établie à l'aide du protocole HLS PUT. Le groupe de sorties contient deux sorties. Une sortie est destinée à la ressource vidéo (nomméeVideo_high) et à la ressource audio (nomméeAudio_EN). L'une des sorties concerne la ressource de sous-titres (nomméeWebVTT_EN).

Dans cet exemple, certains paramètres ne contiennent aucune valeur ou contiennent des paramètres vides imbriqués. Par OutputSettings exemple, car la `Video_and_audio` sortie contient plusieurs paramètres imbriqués qui se terminent par un paramètre vide M3U8Settings. Ce paramètre doit être inclus, mais vous pouvez omettre un, plusieurs ou tous ses enfants, ce qui signifie que l'enfant prendra sa valeur par défaut ou sera nul.

Tous les paramètres qui s'appliquent à cet exemple de canal mais qui ne sont pas spécifiés dans ce fichier prendront la valeur par défaut, seront définis sur null ou prendront une valeur unique générée par MediaLive.

```
aws medialive create-channel \
    --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

Contenu de `channel-in-hls-out-hls-akamai.json` :

```
{
    "Name": "News_West",
    "RoleArn": "arn:aws:iam::111122223333:role/MediaLiveAccessRole",
    "InputAttachments": [
        {
            "InputAttachmentName": "local_news",
            "InputId": "1234567",
            "InputSettings": {
                "AudioSelectors": [
                    {
```

```
                    "Name": "English-Audio",
                    "SelectorSettings": {
                        "AudioLanguageSelection": {
                            "LanguageCode": "EN"
                        }
                    }
                }
            ],
            "CaptionSelectors": [
                {
                    "LanguageCode": "ENE",
                    "Name": "English_embedded"
                }
            ]
        }
    }
],
"Destinations": [
    {
        "Id": "akamai-server-west",
        "Settings": [
            {
                "PasswordParam": "/medialive/examplecorp1",
                "Url": "http://203.0.113.55/news/news_west",
                "Username": "examplecorp"
            },
            {
                "PasswordParam": "/medialive/examplecorp2",
                "Url": "http://203.0.113.82/news/news_west",
                "Username": "examplecorp"
            }
        ]
    }
],
"EncoderSettings": {
    "AudioDescriptions": [
        {
            "AudioSelectorName": "English-Audio",
            "CodecSettings": {
                "AacSettings": {}
            },
            "Name": "Audio_EN"
        }
    ],
```

```
        "CaptionDescriptions": [
            {
                "CaptionSelectorName": "English_embedded",
                "DestinationSettings": {
                    "WebvttDestinationSettings": {}
                },
                "Name": "WebVTT_EN"
            }
        ],
        "VideoDescriptions": [
            {
                "Height": 720,
                "Name": "Video_high",
                "Width": 1280
            }
        ],
        "OutputGroups": [
            {
                "Name": "Akamai",
                "OutputGroupSettings": {
                    "HlsGroupSettings": {
                        "Destination": {
                            "DestinationRefId": "akamai-server-west"
                        },
                        "HlsCdnSettings": {
                            "HlsBasicPutSettings": {}
                        }
                    }
                },
                "Outputs": [
                    {
                        "AudioDescriptionNames": [
                            "Audio_EN"
                        ],
                        "OutputName": "Video_and_audio",
                        "OutputSettings": {
                            "HlsOutputSettings": {
                                "HlsSettings": {
                                    "StandardHlsSettings": {
                                        "M3u8Settings": {}
                                    }
                                },
                                "NameModifier": "_1"
                            }
```

```
                        },
                        "VideoDescriptionName": "Video_high"
                    },
                    {
                        "CaptionDescriptionNames": [
                            "WebVTT_EN"
                        ],
                        "OutputName": "Captions-WebVTT",
                        "OutputSettings": {
                            "HlsOutputSettings": {
                                "HlsSettings": {
                                    "StandardHlsSettings": {
                                        "M3u8Settings": {}
                                    }
                                },
                                "NameModifier": "_2"
                            }
                        }
                    }
                ]
            }
        ],
        "TimecodeConfig": {
            "Source": "EMBEDDED"
        }
    }
}
```

Sortie :

La sortie répète le contenu du fichier JSON, ainsi que les valeurs suivantes. Tous les paramètres sont classés par ordre alphabétique.

ARNpour la chaîne. La dernière partie de l'ARN est l'identifiant unique du canal. `EgressEndpoints`est vide dans cet exemple de canal car il n'est utilisé que pour les entrées PUSH. Lorsqu'il s'applique, il indique les adresses vers MediaLive lesquelles le contenu est envoyé. `OutputGroups`,`Outputs`. Ils indiquent tous les paramètres du groupe de sortie et des sorties, y compris ceux que vous n'avez pas inclus mais qui sont pertinents pour ce canal. Les paramètres peuvent être vides (cela indique peut-être que le paramètre ou la fonctionnalité est désactivé dans cette configuration de canal) ou peuvent afficher la valeur par défaut qui s'appliquera. `LogLevel`est défini sur la valeur par défaut (DÉSACTIVÉ). `Tags`est défini sur la valeur par défaut (null). `PipelinesRunningCount`et `State` affiche l'état actuel de la chaîne.

Pour plus d'informations, consultez la section Création d'une chaîne à partir de zéro dans le guide de l' MediaLive utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateChannelà la section Référence des AWS CLI commandes.

`create-input`

L'exemple de code suivant montre comment utiliser`create-input`.

AWS CLI

Pour créer une entrée

L'`create-input`exemple suivant crée une `HLS PULL` entrée en transmettant un fichier JSON contenant les paramètres qui s'appliquent à ce type d'entrée. Le JSON de cet exemple d'entrée spécifie deux sources (adresses) pour l'entrée, afin de prendre en charge la redondance lors de l'ingestion. Ces adresses nécessitent un mot de passe.

```
aws medialive create-input \
    --cli-input-json file://input-hls-pull-news.json
```

Contenu de `input-hls-pull-news.json` :

```
{
    "Name": "local_news",
    "RequestId": "cli000059",
    "Sources": [
            {
                    "Url": "https://203.0.113.13/newschannel/anytownusa.m3u8",
                    "Username": "examplecorp",
                    "PasswordParam": "/medialive/examplecorp1"
            },
       {
                    "Url": "https://198.51.100.54/fillervideos/oceanwaves.mp4",
                    "Username": "examplecorp",
                    "PasswordParam": "examplecorp2"
            }
    ],
    "Type": "URL_PULL"
}
```

Sortie :

La sortie répète le contenu du fichier JSON, ainsi que les valeurs suivantes. Tous les paramètres sont classés par ordre alphabétique.

`Arn`pour l'entrée. La dernière partie de l'ARN est l'identifiant d'entrée unique. `Attached Channels`, qui est toujours vide pour une entrée nouvellement créée. `Destinations`, qui est vide dans cet exemple car il est utilisé uniquement avec une entrée PUSH. `Id`pour l'entrée, identique à l'ID dans l'ARN. `MediaConnectFlows`, qui est vide dans cet exemple car il est utilisé uniquement avec une entrée de type MediaConnect. `SecurityGroups`, qui est vide dans cet exemple car il est utilisé uniquement avec une entrée PUSH. `State`de cette entrée. `Tags`, qui est vide (valeur par défaut pour ce paramètre).

Pour plus d'informations, consultez la section Création d'une entrée dans le guide de l' MediaLive utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateInputà la section Référence des AWS CLI commandes.

MediaPackage exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with MediaPackage.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-channel`

L'exemple de code suivant montre comment utiliser`create-channel`.

AWS CLI

Pour créer une chaîne

La `create-channel` commande suivante crée un canal nommé `sportschannel` dans le compte courant.

```
aws mediapackage create-channel --id sportschannel
```

Sortie :

```
{
   "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
   "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
                "Password": "generatedwebdavpassword1",
                "Url": "https://f31c86aed53b815a.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
                "Username": "generatedwebdavusername1"
            },
            {
                "Id": "2daa32878af24803b24183727211b8ff",
                "Password": "generatedwebdavpassword2",
                "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
                "Username": "generatedwebdavusername2"
            }
        ]
    },
    "Id": "sportschannel",
    "Tags": {
        "region": "west"
    }
}
```

Pour plus d'informations, consultez la section Création d'une chaîne dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateChannelà la section Référence des AWS CLI commandes.

`create-origin-endpoint`

L'exemple de code suivant montre comment utiliser`create-origin-endpoint`.

AWS CLI

Pour créer un point de terminaison d'origine

La `create-origin-endpoint` commande suivante crée un point de terminaison d'origine nommé `cmafsports` avec les paramètres du package fournis dans un fichier JSON et les paramètres de point de terminaison spécifiés.

```
aws mediapackage create-origin-endpoint \
    --channel-id sportschannel \
    --id cmafsports \
    --cmaf-package file://file/path/cmafpkg.json --description "cmaf output of
 sports" \
    --id cmaf_sports \
    --manifest-name sports_channel \
    --startover-window-seconds 300 \
    --tags region=west,media=sports \
    --time-delay-seconds 10
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
    "ChannelId": "sportschannel",
    "CmafPackage": {
        "HlsManifests": [
            {
                "AdMarkers": "PASSTHROUGH",
                "Id": "cmaf_sports_endpoint",
                "IncludeIframeOnlyStream": true,
                "ManifestName": "index",
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 300,
```

```
                "ProgramDateTimeIntervalSeconds": 300,
                "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/
index.m3u8"
            }
        ],
        "SegmentDurationSeconds": 2,
        "SegmentPrefix": "sportschannel"
    },
    "Description": "cmaf output of sports",
    "Id": "cmaf_sports",
    "ManifestName": "sports_channel",
    "StartoverWindowSeconds": 300,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 10,
    "Url": "",
    "Whitelist": []
}
```

Pour plus d'informations, consultez la section Création d'un point de terminaison dans le guide de MediaPackage l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous CreateOriginEndpointà la section Référence des AWS CLI commandes.

`delete-channel`

L'exemple de code suivant montre comment utiliser`delete-channel`.

AWS CLI

Pour supprimer une chaîne

La `delete-channel` commande suivante supprime le canal nommé`test`.

```
aws mediapackage delete-channel \
    --id test
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'une chaîne dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteChannelà la section Référence des AWS CLI commandes.

`delete-origin-endpoint`

L'exemple de code suivant montre comment utiliser`delete-origin-endpoint`.

AWS CLI

Pour supprimer un point de terminaison d'origine

La `delete-origin-endpoint` commande suivante supprime le point de terminaison d'origine nommé`tester2`.

```
aws mediapackage delete-origin-endpoint \
    --id tester2
```

Pour plus d'informations, consultez la section Suppression d'un point de terminaison dans le guide de MediaPackage l'utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteOriginEndpointà la section Référence des AWS CLI commandes.

`describe-channel`

L'exemple de code suivant montre comment utiliser`describe-channel`.

AWS CLI

Pour décrire une chaîne

La `describe-channel` commande suivante affiche tous les détails du canal nommé`test`.

```
aws mediapackage describe-channel \
    --id test
```

Sortie :

```
{
```

```
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
    "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "584797f1740548c389a273585dd22a63",
                "Password": "webdavgeneratedpassword1",
                "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
                "Username": "webdavgeneratedusername1"
            },
            {
                "Id": "7d187c8616fd455f88aaa5a9fcf74442",
                "Password": "webdavgeneratedpassword2",
                "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
                "Username": "webdavgeneratedusername2"
            }
        ]
    },
    "Id": "test",
    "Tags": {}
}
```

Pour plus d'informations, consultez la section Affichage des détails de la chaîne < https:// docs.aws.amazon.com/mediapackage/ latest/ug/channels -view.html> dans le guide de l'utilisateur élémentaire AWS MediaPackage

- Pour plus de détails sur l'API, reportez-vous DescribeChannelà la section Référence des AWS CLI commandes.

`describe-origin-endpoint`

L'exemple de code suivant montre comment utiliser`describe-origin-endpoint`.

AWS CLI

Pour décrire un point de terminaison d'origine

La `describe-origin-endpoint` commande suivante affiche tous les détails du point de terminaison d'origine nommé`cmaf_sports`.

```
aws mediapackage describe-origin-endpoint \
    --id cmaf_sports
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
    "ChannelId": "sportschannel",
    "CmafPackage": {
        "HlsManifests": [
            {
                "AdMarkers": "NONE",
                "Id": "cmaf_sports_endpoint",
                "IncludeIframeOnlyStream": false,
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 60,
                "ProgramDateTimeIntervalSeconds": 0,
                "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/
index.m3u8"
            }
        ],
        "SegmentDurationSeconds": 2,
        "SegmentPrefix": "sportschannel"
    },
    "Id": "cmaf_sports",
    "ManifestName": "index",
    "StartoverWindowSeconds": 0,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 0,
    "Url": "",
    "Whitelist": []
}
```

Pour plus d'informations, consultez la section Affichage d'un seul point de terminaison dans le guide de MediaPackage l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous DescribeOriginEndpointà la section Référence des AWS CLI commandes.

`list-channels`

L'exemple de code suivant montre comment utiliser`list-channels`.

AWS CLI

Pour répertorier toutes les chaînes

La `list-channels` commande suivante répertorie tous les canaux configurés sur le AWS compte courant.

```
aws mediapackage list-channels
```

Sortie :

```
{
    "Channels": [
        {
            "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
            "HlsIngest": {
                "IngestEndpoints": [
                    {
                        "Id": "584797f1740548c389a273585dd22a63",
                        "Password": "webdavgeneratedpassword1",
                        "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
                        "Username": "webdavgeneratedusername1"
                    },
                    {
                        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
                        "Password": "webdavgeneratedpassword2",
                        "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
                        "Username": "webdavgeneratedusername2"
                    }
                ]
            },
            "Id": "test",
            "Tags": {}
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails des chaînes dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListChannelsà la section Référence des AWS CLI commandes.

`list-origin-endpoints`

L'exemple de code suivant montre comment utiliser`list-origin-endpoints`.

AWS CLI

Pour répertorier tous les points d'origine et de terminaison d'une chaîne

La `list-origin-endpoints` commande suivante répertorie tous les points de terminaison d'origine configurés sur le canal nommé`test`.

```
aws mediapackage list-origin-endpoints \
    --channel-id test
```

Sortie :

```
{
    "OriginEndpoints": [
        {
            "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/247cff871f2845d3805129be22f2c0a2",
            "ChannelId": "test",
            "DashPackage": {
                "ManifestLayout": "FULL",
                "ManifestWindowSeconds": 60,
                "MinBufferTimeSeconds": 30,
                "MinUpdatePeriodSeconds": 15,
                "PeriodTriggers": [],
                "Profile": "NONE",
                "SegmentDurationSeconds": 2,
                "SegmentTemplateFormat": "NUMBER_WITH_TIMELINE",
                "StreamSelection": {
                    "MaxVideoBitsPerSecond": 2147483647,
                    "MinVideoBitsPerSecond": 0,
```

```
                    "StreamOrder": "ORIGINAL"
                },
                "SuggestedPresentationDelaySeconds": 25
            },
            "Id": "tester2",
            "ManifestName": "index",
            "StartoverWindowSeconds": 0,
            "Tags": {},
            "TimeDelaySeconds": 0,
            "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/247cff871f2845d3805129be22f2c0a2/index.mpd",
            "Whitelist": []
        },
        {
            "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/869e237f851549e9bcf10e3bc2830839",
            "ChannelId": "test",
            "HlsPackage": {
                "AdMarkers": "NONE",
                "IncludeIframeOnlyStream": false,
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 60,
                "ProgramDateTimeIntervalSeconds": 0,
                "SegmentDurationSeconds": 6,
                "StreamSelection": {
                    "MaxVideoBitsPerSecond": 2147483647,
                    "MinVideoBitsPerSecond": 0,
                    "StreamOrder": "ORIGINAL"
                },
                "UseAudioRenditionGroup": false
            },
            "Id": "tester",
            "ManifestName": "index",
            "StartoverWindowSeconds": 0,
            "Tags": {},
            "TimeDelaySeconds": 0,
            "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/869e237f851549e9bcf10e3bc2830839/index.m3u8",
            "Whitelist": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage de tous les points de terminaison associés à un canal dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListOriginEndpointsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises attribuées à une ressource

La `list-tags-for-resource` commande suivante répertorie les balises attribuées à la ressource spécifiée.

```
aws mediapackage list-tags-for-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0
```

Sortie :

```
{
    "Tags": {
        "region": "west"
    }
}
```

Pour plus d'informations, consultez la section Marquage des ressources dans AWS Elemental MediaPackage dans le guide de l'utilisateur d'AWS Elemental MediaPackage .

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`rotate-ingest-endpoint-credentials`

L'exemple de code suivant montre comment utiliser`rotate-ingest-endpoint-credentials`.

AWS CLI

Pour faire pivoter les informations d'ingestion

La `rotate-ingest-endpoint-credentials` commande suivante fait pivoter le nom d'utilisateur et le mot de passe WebDAV pour le point de terminaison d'ingestion spécifié.

```
aws mediapackage rotate-ingest-endpoint-credentials \
    --id test \
    --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
    "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "584797f1740548c389a273585dd22a63",
                "Password": "webdavregeneratedpassword1",
                "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
                "Username": "webdavregeneratedusername1"
            },
            {
                "Id": "7d187c8616fd455f88aaa5a9fcf74442",
                "Password": "webdavgeneratedpassword2",
                "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
                "Username": "webdavgeneratedusername2"
            }
        ]
    },
    "Id": "test",
    "Tags": {}
}
```

Pour plus d'informations, voir Rotation des informations d'identification sur une URL d'entrée dans le Guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous RotateIngestEndpointCredentialsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Ajout d'une balise à une ressource

Les `tag-resource` commandes suivantes ajoutent une paire `region=west` clé/valeur à la ressource spécifiée.

```
aws mediapackage tag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tags region=west
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources dans AWS Elemental MediaPackage dans le guide de l'utilisateur d'AWS Elemental MediaPackage .

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'une ressource

La `untag-resource` commande suivante supprime le tag contenant la clé `region` du canal spécifié.

```
aws mediapackage untag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tag-keys region
```

Pour plus d'informations, consultez la section Marquage des ressources dans AWS Elemental MediaPackage dans le guide de l'utilisateur d'AWS Elemental MediaPackage .

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-channel`

L'exemple de code suivant montre comment utiliser`update-channel`.

AWS CLI

Pour mettre à jour une chaîne

La `update-channel` commande suivante met à jour le canal nommé `sportschannel` pour inclure la description`24x7 sports`.

```
aws mediapackage update-channel \
    --id sportschannel \
    --description "24x7 sports"
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
    "Description": "24x7 sports",
    "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
                "Password": "generatedwebdavpassword1",
                "Url": "https://f31c86aed53b815a.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
                "Username": "generatedwebdavusername1"
            },
            {
                "Id": "2daa32878af24803b24183727211b8ff",
                "Password": "generatedwebdavpassword2",
                "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
                "Username": "generatedwebdavusername2"
            }
```

```
        ]
    },
    "Id": "sportschannel",
    "Tags": {}
}
```

Pour plus d'informations, consultez la section Modification d'une chaîne dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous UpdateChannelà la section Référence des AWS CLI commandes.

`update-origin-endpoint`

L'exemple de code suivant montre comment utiliser`update-origin-endpoint`.

AWS CLI

Pour mettre à jour un point de terminaison d'origine

La `update-origin-endpoint` commande suivante met à jour le point de terminaison d'origine nommé`cmaf_sports`. Cela change le délai en `0` secondes.

```
aws mediapackage update-origin-endpoint \
    --id cmaf_sports \
    --time-delay-seconds 0
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
    "ChannelId": "sportschannel",
    "CmafPackage": {
        "HlsManifests": [
            {
                "AdMarkers": "NONE",
                "Id": "cmaf_sports_endpoint",
                "IncludeIframeOnlyStream": false,
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 60,
```

```
                "ProgramDateTimeIntervalSeconds": 0,
                "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/
index.m3u8"
            }
        ],
        "SegmentDurationSeconds": 2,
        "SegmentPrefix": "sportschannel"
    },
    "Id": "cmaf_sports",
    "ManifestName": "index",
    "StartoverWindowSeconds": 0,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 0,
    "Url": "",
    "Whitelist": []
}
```

Pour plus d'informations, consultez la section Modification d'un point de terminaison dans le guide de MediaPackage l'utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous UpdateOriginEndpointà la section Référence des AWS CLI commandes.

MediaPackage Exemples de VOD utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de la MediaPackage VOD.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-asset`

L'exemple de code suivant montre comment utiliser`create-asset`.

AWS CLI

Pour créer un actif

L'`create-asset`exemple suivant crée un actif nommé `Chicken_Asset` dans le AWS compte courant. La ressource ingère le fichier dans`30sec_chicken.smil`. MediaPackage

```
aws mediapackage-vod create-asset \
    --id chicken_asset \
    --packaging-group-id hls_chicken_gp \
    --source-role-arn arn:aws:iam::111122223333:role/EMP_Vod \
    --source-arn arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil
```

Sortie :

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:assets/chicken_asset",
    "Id":"chicken_asset",
    "PackagingGroupId":"hls_chicken_gp",
    "SourceArn":"arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
    "SourceRoleArn":"arn:aws:iam::111122223333:role/EMP_Vod",
    "EgressEndpoints":[
        {
            "PackagingConfigurationId":"New_config_1",
            "Url":"https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/
v1/6644b55df1744261ab3732a8e5cdaf07/904b06a58c7645e08d57d40d064216ac/
f5b2e633ff4942228095d164c10074f3/index.m3u8"
        },
        {
            "PackagingConfigurationId":"new_hls",
            "Url":" https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/6644b55df1744261ab3732a8e5cdaf07/
fe8f1f00a80e424cb4f8da4095835e9e/7370ec57432343af816332356d2bd5c6/string.m3u8"
        }
    ]
}
```

Pour plus d'informations, voir Ingérer un actif dans le guide de l' MediaPackage utilisateur d'AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreateAssetà la section Référence des AWS CLI commandes.

`create-packaging-configuration`

L'exemple de code suivant montre comment utiliser`create-packaging-configuration`.

AWS CLI

Pour créer une configuration d'emballage

L'`create-packaging-configuration`exemple suivant crée une configuration d'emballage nommée `new_hls` dans le groupe d'emballage nommé`hls_chicken`. Cet exemple utilise un fichier sur le disque nommé `hls_pc.json` pour fournir les détails.

```
aws mediapackage-vod create-packaging-configuration \
    --id new_hls \
    --packaging-group-id hls_chicken \
    --hls-package file://hls_pc.json
```

Contenu de `hls_pc.json` :

```
{
    "HlsManifests":[
        {
            "AdMarkers":"NONE",
            "IncludeIframeOnlyStream":false,
            "ManifestName":"string",
            "ProgramDateTimeIntervalSeconds":60,
            "RepeatExtXKey":true,
            "StreamSelection":{
                "MaxVideoBitsPerSecond":1000,
                "MinVideoBitsPerSecond":0,
                "StreamOrder":"ORIGINAL"
            }
        }
    ],
    "SegmentDurationSeconds":6,
    "UseAudioRenditionGroup":false
```

```
}
```

Sortie :

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/
new_hls",
    "Id":"new_hls",
    "PackagingGroupId":"hls_chicken",
    "HlsManifests":{
        "SegmentDurationSeconds":6,
        "UseAudioRenditionGroup":false,
        "HlsMarkers":[
            {
                "AdMarkers":"NONE",
                "IncludeIframeOnlyStream":false,
                "ManifestName":"string",
                "ProgramDateTimeIntervalSeconds":60,
                "RepeatExtXKey":true,
                "StreamSelection":{
                    "MaxVideoBitsPerSecond":1000,
                    "MinVideoBitsPerSecond":0,
                    "StreamOrder":"ORIGINAL"
                }
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Création d'une configuration d'emballage dans le Guide de MediaPackage l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous CreatePackagingConfigurationà la section Référence des AWS CLI commandes.

`create-packaging-group`

L'exemple de code suivant montre comment utiliser`create-packaging-group`.

AWS CLI

Pour créer un groupe d'emballages

L'`create-packaging-group`exemple suivant répertorie tous les groupes de packaging configurés dans le AWS compte courant.

```
aws mediapackage-vod create-packaging-group \
    --id hls_chicken
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
hls_chicken",
    "Id": "hls_chicken"
}
```

Pour plus d'informations, consultez la section Création d'un groupe de packaging dans le Guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous CreatePackagingGroupà la section Référence des AWS CLI commandes.

`delete-asset`

L'exemple de code suivant montre comment utiliser`delete-asset`.

AWS CLI

Pour supprimer un actif

L'`delete-asset`exemple suivant supprime l'actif nommé`30sec_chicken`.

```
aws mediapackage-vod delete-asset \
    --id 30sec_chicken
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un actif dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteAssetà la section Référence des AWS CLI commandes.

`delete-packaging-configuration`

L'exemple de code suivant montre comment utiliser`delete-packaging-configuration`.

AWS CLI

Pour supprimer une configuration d'emballage

L'`delete-packaging-configuration`exemple suivant supprime la configuration d'emballage nomméeCMAF.

```
aws mediapackage-vod delete-packaging-configuration \
    --id CMAF
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'une configuration d'emballage dans le guide de MediaPackage l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous DeletePackagingConfigurationà la section Référence des AWS CLI commandes.

`delete-packaging-group`

L'exemple de code suivant montre comment utiliser`delete-packaging-group`.

AWS CLI

Pour supprimer un groupe d'emballages

L'`delete-packaging-group`exemple suivant supprime le groupe d'emballage nomméDash_widevine.

```
aws mediapackage-vod delete-packaging-group \
    --id Dash_widevine
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un groupe de packaging dans le guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeletePackagingGroupà la section Référence des AWS CLI commandes.

`describe-asset`

L'exemple de code suivant montre comment utiliser`describe-asset`.

AWS CLI

Pour décrire un actif

L'`describe-asset`exemple suivant affiche tous les détails de la ressource nommée`30sec_chicken`.

```
aws mediapackage-vod describe-asset \
    --id 30sec_chicken
```

Sortie :

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:assets/30sec_chicken",
    "Id":"30sec_chicken",
    "PackagingGroupId":"Packaging_group_1",
    "SourceArn":"arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
    "SourceRoleArn":"arn:aws:iam::111122223333:role/EMP_Vod",
    "EgressEndpoints":[
        {
            "PackagingConfigurationId":"DASH",
            "Url":"https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/66c25aff456d463aae0855172b3beb27/4ddfda6da17c4c279a1b8401cb
index.mpd"
        },
        {
            "PackagingConfigurationId":"HLS",
            "Url":"https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/6e5bf286a3414254a2bf0d22ae148d7e/06b5875b4d004c3cbdc4da2dc4
index.m3u8"
        },
        {
            "PackagingConfigurationId":"CMAF",
```

```
            "Url":"https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/628fb5d8d89e4702958b020af27fde0e/05eb062214064238ad6330a443
index.m3u8"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails des actifs dans le Guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribeAsseten à la section Référence des AWS CLI commandes.

`describe-packaging-configuration`

L'exemple de code suivant montre comment utiliser`describe-packaging-configuration`.

AWS CLI

Pour décrire une configuration d'emballage

L'`describe-packaging-configuration`exemple suivant affiche tous les détails de la configuration d'emballage nomméeDASH.

```
aws mediapackage-vod describe-packaging-configuration \
    --id DASH
```

Sortie :

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/
DASH",
    "Id":"DASH",
    "PackagingGroupId":"Packaging_group_1",
    "DashPackage":[
        {
            "SegmentDurationSeconds":"2"
        },
        {
            "DashManifests":{
```

```
                "ManifestName":"index",
                "MinBufferTimeSeconds":"30",
                "Profile":"NONE"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails de configuration des packages dans le Guide de MediaPackage l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous DescribePackagingConfigurationà la section Référence des AWS CLI commandes.

`describe-packaging-group`

L'exemple de code suivant montre comment utiliser`describe-packaging-group`.

AWS CLI

Pour décrire un groupe d'emballages

L'`describe-packaging-group`exemple suivant affiche tous les détails du groupe d'emballage nomméPackaging_group_1.

```
aws mediapackage-vod describe-packaging-group \
    --id Packaging_group_1
```

Sortie :

```
{
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
Packaging_group_1",
    "Id": "Packaging_group_1"
}
```

Pour plus d'informations, consultez la section Affichage des détails des groupes de packaging dans le Guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribePackagingGroupà la section Référence des AWS CLI commandes.

`list-assets`

L'exemple de code suivant montre comment utiliser`list-assets`.

AWS CLI

Pour répertorier tous les actifs

L'`list-assets`exemple suivant répertorie tous les actifs configurés dans le AWS compte courant.

```
aws mediapackage-vod list-assets
```

Sortie :

```
{
    "Assets": [
        "Arn": "arn:aws:mediapackage-vod:us-
west-2:111122223333:assets/30sec_chicken",
        "Id": "30sec_chicken",
        "PackagingGroupId": "Packaging_group_1",
        "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
        "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod"
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails des actifs dans le Guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListAssetsà la section Référence des AWS CLI commandes.

`list-packaging-configurations`

L'exemple de code suivant montre comment utiliser`list-packaging-configurations`.

AWS CLI

Pour répertorier toutes les configurations d'emballage

L'`list-packaging-configurations`exemple suivant répertorie toutes les configurations d'emballage configurées sur le groupe d'emballage nomméPackaging_group_1.

```
aws mediapackage-vod list-packaging-configurations \
    --packaging-group-id Packaging_group_1
```

Sortie :

```
{
    "PackagingConfigurations":[
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/CMAF",
            "Id":"CMAF",
            "PackagingGroupId":"Packaging_group_1",
            "CmafPackage":[
                {
                    "SegmentDurationSeconds":"2"
                },
                {
                    "HlsManifests":{
                        "AdMarkers":"NONE",
                        "RepeatExtXKey":"False",
                        "ManifestName":"index",
                        "ProgramDateTimeIntervalSeconds":"0",
                        "IncludeIframeOnlyStream":"False"
                    }
                }
            ]
        },
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/DASH",
            "Id":"DASH",
            "PackagingGroupId":"Packaging_group_1",
            "DashPackage":[
                {
                    "SegmentDurationSeconds":"2"
                },
                {
                    "DashManifests":{
                        "ManifestName":"index",
                        "MinBufferTimeSeconds":"30",
                        "Profile":"NONE"
                    }
                }
```

```
            ]
        },
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/HLS",
            "Id":"HLS",
            "PackagingGroupId":"Packaging_group_1",
            "HlsPackage":[
                {
                    "SegmentDurationSeconds":"6",
                    "UseAudioRenditionGroup":"False"
                },
                {
                    "HlsManifests":{
                        "AdMarkers":"NONE",
                        "RepeatExtXKey":"False",
                        "ManifestName":"index",
                        "ProgramDateTimeIntervalSeconds":"0",
                        "IncludeIframeOnlyStream":"False"
                    }
                }
            ]
        },
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/New_config_0_copy",
            "Id":"New_config_0_copy",
            "PackagingGroupId":"Packaging_group_1",
            "HlsPackage":[
                {
                    "SegmentDurationSeconds":"6",
                    "UseAudioRenditionGroup":"False"
                },
                {
                    "Encryption":{
                        "EncryptionMethod":"AWS_128",
                        "SpekeKeyProvider":{
                           "RoleArn":"arn:aws:iam:111122223333::role/SPEKERole",
                            "Url":"https://lfgubdvs97.execute-api.us-
west-2.amazonaws.com/EkeStage/copyProtection/",
                            "SystemIds":[
                                "81376844-f976-481e-a84e-cc25d39b0b33"
                            ]
                        }
```

```
                    }
                },
                {
                    "HlsManifests":{
                        "AdMarkers":"NONE",
                        "RepeatExtXKey":"False",
                        "ManifestName":"index",
                        "ProgramDateTimeIntervalSeconds":"0",
                        "IncludeIframeOnlyStream":"False"
                    }
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails de configuration des packages dans le Guide de MediaPackage l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous ListPackagingConfigurationsà la section Référence des AWS CLI commandes.

`list-packaging-groups`

L'exemple de code suivant montre comment utiliser`list-packaging-groups`.

AWS CLI

Pour répertorier tous les groupes d'emballages

L'`list-packaging-groups`exemple suivant répertorie tous les groupes de packaging configurés dans le AWS compte courant.

```
aws mediapackage-vod list-packaging-groups
```

Sortie :

```
{
    "PackagingGroups": [
        {
            "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Dash_widevine",
```

```
            "Id": "Dash_widevine"
        },
        {
            "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Encrypted_HLS",
            "Id": "Encrypted_HLS"
        },
        {
            "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Packaging_group_1",
            "Id": "Packaging_group_1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails des groupes de packaging dans le Guide de l' MediaPackage utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListPackagingGroupsà la section Référence des AWS CLI commandes.

MediaStore Exemples de plans de données utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du plan de MediaStore données AWS Command Line Interface with.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-object`

L'exemple de code suivant montre comment utiliser`delete-object`.

AWS CLI

Pour supprimer un objet

L'`delete-object`exemple suivant supprime l'objet spécifié.

```
aws mediastore-data delete-object \
    --endpoint=https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS CLI commandes.

`describe-object`

L'exemple de code suivant montre comment utiliser`describe-object`.

AWS CLI

Pour afficher les en-têtes d'un objet

L'`describe-object`exemple suivant affiche les en-têtes d'un objet sur le chemin spécifié.

```
aws mediastore-data describe-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg
```

Sortie :

```
{
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
    "ContentType": "image/jpeg",
    "ContentLength": "3860266",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Pour plus d'informations, consultez la section Affichage des détails d'un objet dans le Guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous DescribeObjectà la section Référence des AWS CLI commandes.

`get-object`

L'exemple de code suivant montre comment utiliser`get-object`.

AWS CLI

Exemple 1 : pour télécharger un objet entier

L'`get-object`exemple suivant télécharge l'objet spécifié.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg setup.jpg
```

Sortie :

```
{
    "ContentType": "image/jpeg",
    "StatusCode": 200,
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentLength": "3860266",
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"
}
```

Exemple 2 : pour télécharger une partie d'un objet

L'`get-object`exemple suivant télécharge la partie spécifiée d'un objet.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg setup.jpg \
    --range "bytes=0-100"
```

Sortie :

```
{
    "StatusCode": 206,
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
    "ContentType": "image/jpeg",
    "ContentRange": "bytes 0-100/3860266",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentLength": "101"
}
```

Pour plus d'informations, consultez la section Téléchargement d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

`list-items`

L'exemple de code suivant montre comment utiliser`list-items`.

AWS CLI

Exemple 1 : pour afficher la liste des éléments (objets et dossiers) stockés dans un conteneur

L'`list-items`exemple suivant affiche la liste des éléments (objets et dossiers) stockés dans le conteneur spécifié.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

Sortie :

```
{
    "Items": [
        {
            "Type": "OBJECT",
            "ContentLength": 3784,
            "Name": "setup.jpg",
            "ETag":
 "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379
```

```
        },
        {
            "Type": "FOLDER",
            "Name": "events"
        }
    ]
}
```

Exemple 2 : pour afficher la liste des éléments (objets et dossiers) stockés dans un dossier

L'`list-items`exemple suivant affiche la liste des éléments (objets et dossiers) stockés dans le dossier spécifié.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball
```

Sortie :

```
{
    "Items": [
        {
            "ETag":
 "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
            "ContentType": "image/jpeg",
            "Type": "OBJECT",
            "ContentLength": 3860266,
            "LastModified": 1563573031.872,
            "Name": "setup.jpg"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une liste d'objets dans le Guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous ListItemsà la section Référence des AWS CLI commandes.

`put-object`

L'exemple de code suivant montre comment utiliser`put-object`.

AWS CLI

Exemple 1 : pour télécharger un objet dans un conteneur

L'`put-object`exemple suivant télécharge un objet dans le conteneur spécifié.

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body ReadMe.md \
    --path ReadMe.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

Sortie :

```
{
    "ContentSHA256":
 "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

Exemple 2 : pour télécharger un objet dans un dossier au sein d'un conteneur

L'`put-object`exemple suivant télécharge un objet dans le dossier spécifié au sein d'un conteneur.

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body ReadMe.md \
    --path /september-events/ReadMe.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

Sortie :

```
{
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentSHA256":
 "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL"
}
```

Pour plus d'informations, consultez la section Chargement d'un objet dans le guide de l' MediaStore utilisateur AWS Elemental.

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

MediaTailor exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with MediaTailor.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-playback-configuration`

L'exemple de code suivant montre comment utiliser`delete-playback-configuration`.

AWS CLI

Pour supprimer une configuration

Ce qui suit `delete-playback-configuration` supprime une configuration nomméecampaign_short.

```
aws mediatailor delete-playback-configuration \
    --name campaign_short
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'une configuration dans le Guide de MediaTailor l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous DeletePlaybackConfigurationà la section Référence des AWS CLI commandes.

`get-playback-configuration`

L'exemple de code suivant montre comment utiliser`get-playback-configuration`.

AWS CLI

Pour décrire une configuration

Ce qui suit `get-playback-configuration` affiche tous les détails de la configuration nommée`west_campaign`.

```
aws mediatailor get-playback-configuration \
    --name west_campaign
```

Sortie :

```
{
     "AdDecisionServerUrl": "http://your.ads.url",
     "CdnConfiguration": {},
     "DashConfiguration": {
         "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
         "MpdLocation": "EMT_DEFAULT",
         "OriginManifestType": "MULTI_PERIOD"
     },
     "HlsConfiguration": {
         "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
     },
     "Name": "west_campaign",
     "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/west_campaign",
     "PlaybackEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
     "SessionInitializationEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
```

```
    "Tags": {},
    "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-
west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
}
```

Pour plus d'informations, consultez la section Affichage d'une configuration dans le Guide de MediaTailor l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous GetPlaybackConfigurationà la section Référence des AWS CLI commandes.

`list-playback-configurations`

L'exemple de code suivant montre comment utiliser`list-playback-configurations`.

AWS CLI

Pour répertorier toutes les configurations

Ce qui suit `list-playback-configurations` affiche tous les détails de la configuration sur le AWS compte courant.

```
aws mediatailor list-playback-configurations
```

Sortie :

```
{
    "Items": [
        {
            "AdDecisionServerUrl": "http://your.ads.url",
            "CdnConfiguration": {},
            "DashConfiguration": {
                "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
                "MpdLocation": "EMT_DEFAULT",
                "OriginManifestType": "MULTI_PERIOD"
            },
            "HlsConfiguration": {
                "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
            },
```

```
            "Name": "west_campaign",
            "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/west_campaign",
            "PlaybackEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
            "SessionInitializationEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
            "Tags": {},
            "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-
west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
        },
        {
            "AdDecisionServerUrl": "http://your.ads.url",
            "CdnConfiguration": {},
            "DashConfiguration": {
                "ManifestEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
                "MpdLocation": "DISABLED",
                "OriginManifestType": "MULTI_PERIOD"
            },
            "HlsConfiguration": {
                "ManifestEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/"
            },
            "Name": "sports_campaign",
            "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/sports_campaign",
            "PlaybackEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com",
            "SessionInitializationEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
            "SlateAdUrl": "http://s3.bucket/slate_ad.mp4",
            "Tags": {},
            "VideoContentSourceUrl": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/sports_endpoint/
index.m3u8"
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage d'une configuration < https://docs.aws.amazon.com/mediatailor/ latest/ug/configurations -view.html> dans le Guide de l'utilisateur élémentaire.AWS MediaTailor

- Pour plus de détails sur l'API, reportez-vous ListPlaybackConfigurationsà la section Référence des AWS CLI commandes.

`put-playback-configuration`

L'exemple de code suivant montre comment utiliser`put-playback-configuration`.

AWS CLI

Pour créer une configuration

Ce qui suit `put-playback-configuration` crée une configuration nomméecampaign_short.

```
aws mediatailor put-playback-configuration \
    --name campaign_short \
    --ad-decision-server-url http://your.ads.url \
    --video-content-source-url http://video.bucket/index.m3u8
```

Sortie :

```
{
    "AdDecisionServerUrl": "http://your.ads.url",
    "CdnConfiguration": {},
    "DashConfiguration": {
         "ManifestEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",
         "MpdLocation": "EMT_DEFAULT",
         "OriginManifestType": "MULTI_PERIOD"
    },
    "HlsConfiguration": {
        "ManifestEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/"
    },
    "Name": "campaign_short",
    "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/campaign_short",
```

```
    "PlaybackEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com",
    "SessionInitializationEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",
    "Tags": {},
    "VideoContentSourceUrl": "http://video.bucket/index.m3u8"
}
```

Pour plus d'informations, consultez la section Création d'une configuration dans le Guide de MediaTailor l'utilisateur AWS élémentaire.

- Pour plus de détails sur l'API, reportez-vous PutPlaybackConfigurationà la section Référence des AWS CLI commandes.

Exemples de MemoryDB utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with MemoryDB.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`copy-snapshot`

L'exemple de code suivant montre comment utiliser`copy-snapshot`.

AWS CLI

Pour copier un instantané

L'`copy-snapshot`exemple suivant crée une copie d'un instantané.

```
aws memorydb copy-snapshot \
    --source-snapshot-name my-cluster-snapshot \
    --target-snapshot-name my-cluster-snapshot-copy
```

Sortie

```
{
    "Snapshot": {
        "Name": "my-cluster-snapshot-copy",
        "Status": "creating",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:snapshot/my-cluster-
snapshot-copy",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": " ",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-xx2574fc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
}
```

Pour plus d'informations, consultez la section Copier un instantané dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CopySnapshotà la section Référence des AWS CLI commandes.

`create-acl`

L'exemple de code suivant montre comment utiliser`create-acl`.

AWS CLI

Pour créer une ACL

L'`create-acl`exemple suivant crée une nouvelle liste de contrôle d'accès.

```
aws memorydb create-acl \
    --acl-name "new-acl-1" \
    --user-names "my-user"
```

Sortie :

```
{
    "ACL": {
        "Name": "new-acl-1",
        "Status": "creating",
        "UserNames": [
            "my-user"
        ],
        "MinimumEngineVersion": "6.2",
        "Clusters": [],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"
    }
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CreateAclà la section Référence des AWS CLI commandes.

`create-cluster`

L'exemple de code suivant montre comment utiliser`create-cluster`.

AWS CLI

Pour créer un cluster

L'`create-cluster`exemple suivant crée un nouveau cluster.

```
aws memorydb create-cluster \
```

```
    --cluster-name my-new-cluster \
    --node-type db.r6g.large \
    --acl-name my-acl \
    --subnet-group my-sg
```

Sortie :

```
{
    "Cluster": {
        "Name": "my-new-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-new-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "sat:10:00-sat:11:00",
        "SnapshotWindow": "07:30-08:30",
        "ACLName": "my-acl",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour plus d'informations, consultez la section Gestion des clusters dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CreateClusterà la section Référence des AWS CLI commandes.

`create-parameter-group`

L'exemple de code suivant montre comment utiliser`create-parameter-group`.

AWS CLI

Pour créer un groupe de paramètres

L'`create-parameter-group`exemple suivant crée un groupe de paramètres.

```
aws memorydb create-parameter-group \
    --parameter-group-name myRedis6x \
    --family memorydb_redis6 \
    --description "my-parameter-group"
```

Sortie :

```
{
    "ParameterGroup": {
        "Name": "myredis6x",
        "Family": "memorydb_redis6",
        "Description": "my-parameter-group",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de paramètres dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CreateParameterGroupà la section Référence des AWS CLI commandes.

`create-snapshot`

L'exemple de code suivant montre comment utiliser`create-snapshot`.

AWS CLI

Pour créer un instantané

L'`create-snapshot`exemple suivant crée un instantané.

```
aws memorydb create-snapshot \
    --cluster-name my-cluster \
    --snapshot-name my-cluster-snapshot
```

Sortie :

```
{
    "Snapshot": {
        "Name": "my-cluster-snapshot1",
        "Status": "creating",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-
snapshot",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": "",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-862xxxxc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
}
```

Pour plus d'informations, consultez la section Création d'instantanés manuels dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CreateSnapshotà la section Référence des AWS CLI commandes.

`create-subnet-group`

L'exemple de code suivant montre comment utiliser`create-subnet-group`.

AWS CLI

Pour créer un groupe de sous-réseaux

L'`create-subnet-group`exemple suivant crée un groupe de sous-réseaux.

```
aws memorydb create-subnet-group \
```

```
    --subnet-group-name mysubnetgroup \
    --description "my subnet group" \
    --subnet-ids subnet-5623xxxx
```

Sortie :

```
{
    "SubnetGroup": {
        "Name": "mysubnetgroup",
        "Description": "my subnet group",
        "VpcId": "vpc-86257xxx",
        "Subnets": [
            {
                "Identifier": "subnet-5623xxxx",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de sous-réseaux dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CreateSubnetGroupà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un utilisateur

L'`create-user`exemple suivant crée un nouvel utilisateur.

```
aws memorydb create-user \
    --user-name user-name-1 \
    --access-string "~objects:* ~items:* ~public:*" \
```

```
    --authentication-mode \
        Passwords="enterapasswordhere",Type=password
```

Sortie :

```
{
    "User": {
        "Name": "user-name-1",
        "Status": "active",
        "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
        "ACLNames": [],
        "MinimumEngineVersion": "6.2",
        "Authentication": {
            "Type": "password",
            "PasswordCount": 1
        },
        "ARN": "arn:aws:memorydb:us-west-2:491658xxxxxx:user/user-name-1"
    }
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`delete-acl`

L'exemple de code suivant montre comment utiliser`delete-acl`.

AWS CLI

Pour supprimer une ACL

L'`delete-acl`exemple suivant supprime une liste de contrôle d'accès.

```
aws memorydb delete-acl \
    --acl-name "new-acl-1"
```

Sortie :

```
{
```

```
    "ACL": {
        "Name": "new-acl-1",
        "Status": "deleting",
        "UserNames": [
            "pat"
        ],
        "MinimumEngineVersion": "6.2",
        "Clusters": [],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"
    }
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DeleteAclà la section Référence des AWS CLI commandes.

`delete-cluster`

L'exemple de code suivant montre comment utiliser`delete-cluster`.

AWS CLI

Pour supprimer un cluster

L'`delete-cluster`exemple suivant supprime un cluster.

```
aws memorydb delete-cluster \
    --cluster-name my-new-cluster
```

Sortie :

```
{
    "Cluster": {
        "Name": "my-new-cluster",
        "Status": "deleting",
        "NumberOfShards": 1,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-new-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
```

```
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-new-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "sat:10:00-sat:11:00",
        "SnapshotWindow": "07:30-08:30",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un cluster dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DeleteClusterà la section Référence des AWS CLI commandes.

`delete-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-parameter-group`.

AWS CLI

Pour supprimer un groupe de paramètres

L'`delete-parameter-group`exemple suivant supprime un groupe de paramètres.

```
aws memorydb delete-parameter-group \
    --parameter-group-name myRedis6x
```

Sortie :

```
{
    "ParameterGroup": {
        "Name": "myredis6x",
        "Family": "memorydb_redis6",
        "Description": "my-parameter-group",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"
```

```
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un groupe de paramètres dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DeleteParameterGroupà la section Référence des AWS CLI commandes.

`delete-snapshot`

L'exemple de code suivant montre comment utiliser`delete-snapshot`.

AWS CLI

Suppression d'un instantané

L'`delete-snapshot`exemple suivant supprime un instantané.

```
aws memorydb delete-snapshot \
    --snapshot-name my-cluster-snapshot
```

Sortie :

```
{
    "Snapshot": {
        "Name": "my-cluster-snapshot",
        "Status": "deleting",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-
snapshot",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": "",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-862xxxxc",
            "SnapshotRetentionLimit": 0,
```

```
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un instantané dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DeleteSnapshotà la section Référence des AWS CLI commandes.

`delete-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-subnet-group`.

AWS CLI

Pour supprimer un groupe de sous-réseaux

L'`delete-subnet-group`exemple suivant supprime un sous-réseau.

```
aws memorydb delete-subnet-group \
    --subnet-group-name mysubnetgroup
```

Sortie :

```
{
    "SubnetGroup": {
        "Name": "mysubnetgroup",
        "Description": "my subnet group",
        "VpcId": "vpc-86xxxx4fc",
        "Subnets": [
            {
                "Identifier": "subnet-56xxx61b",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
    }
```

```
}
```

Pour plus d'informations, consultez la section Suppression d'un groupe de sous-réseaux dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DeleteSubnetGroupà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

L'`delete-user`exemple suivant supprime un utilisateur.

```
aws memorydb delete-user \
    --user-name my-user
```

Sortie :

```
{
    "User": {
        "Name": "my-user",
        "Status": "deleting",
        "AccessString": "on ~app::* resetchannels -@all +@read",
        "ACLNames": [
            "my-acl"
        ],
        "MinimumEngineVersion": "6.2",
        "Authentication": {
            "Type": "password",
            "PasswordCount": 1
        },
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
    }
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`describe-acls`

L'exemple de code suivant montre comment utiliser`describe-acls`.

AWS CLI

Pour renvoyer une liste de ACLs

La commande describe-acls` suivante renvoie une liste de. ACLs

```
aws memorydb describe-acls
```

Sortie :

```
{
    "ACLs": [
        {
            "Name": "open-access",
            "Status": "active",
            "UserNames": [
                "default"
            ],
            "MinimumEngineVersion": "6.2",
            "Clusters": [],
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/open-access"
        },
        {
            "Name": my-acl",
            "Status": "active",
            "UserNames": [],
            "MinimumEngineVersion": "6.2",
            "Clusters": [
                "my-cluster"
            ],
            "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxxx:acl/my-acl"
        }
    ]
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeAclsà la section Référence des AWS CLI commandes.

`describe-clusters`

L'exemple de code suivant montre comment utiliser`describe-clusters`.

AWS CLI

Pour renvoyer une liste de clusters

La commande describe-clusters` suivante renvoie une liste de clusters.

```
aws memorydb describe-clusters
```

Sortie :

```
{
    "Clusters": [
        {
                "Name": "my-cluster",
                "Status": "available",
                "NumberOfShards": 2,
                "ClusterEndpoint": {
                    "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
                    "Port": 6379
                },
                "NodeType": "db.r6g.large",
                "EngineVersion": "6.2",
                "EnginePatchVersion": "6.2.6",
                "ParameterGroupName": "default.memorydb-redis6",
                "ParameterGroupStatus": "in-sync",
                "SecurityGroups": [
                    {
                        "SecurityGroupId": "sg-0a1434xxxxxc9fae",
                        "Status": "active"
                    }
                ],
```

```
                "SubnetGroupName": "pat-sg",
                "TLSEnabled": true,
                "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-cluster",
                "SnapshotRetentionLimit": 0,
                "MaintenanceWindow": "wed:03:00-wed:04:00",
                "SnapshotWindow": "04:30-05:30",
                "ACLName": "my-acl",
                "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des clusters dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeClustersà la section Référence des AWS CLI commandes.

`describe-engine-versions`

L'exemple de code suivant montre comment utiliser`describe-engine-versions`.

AWS CLI

Pour renvoyer une liste des versions du moteur

Le describe-engine-versions `suivant renvoie une liste des versions du moteur.

```
aws memorydb describe-engine-versions
```

Sortie :

```
{
    "EngineVersions": [
        {
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupFamily": "memorydb_redis6"
        }
    ]
}
```

Pour plus d'informations, consultez la section Versions du moteur et mise à niveau dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeEngineVersionsà la section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour renvoyer une liste d'événements

La commande describe-events` suivante renvoie une liste d'événements.

```
aws memorydb describe-events
```

Sortie :

```
{
    "Events": [
        {
            "SourceName": "my-cluster",
            "SourceType": "cluster",
            "Message": "Increase replica count started for replication group my-
cluster on 2022-07-22T14:09:01.440Z",
            "Date": "2022-07-22T07:09:01.443000-07:00"
        },
        {
            "SourceName": "my-user",
            "SourceType": "user",
            "Message": "Create user my-user operation completed.",
            "Date": "2022-07-22T07:00:02.975000-07:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Surveillance des événements dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeEventsà la section Référence des AWS CLI commandes.

`describe-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-parameter-groups`.

AWS CLI

Pour renvoyer une liste de groupes de paramètres

Le describe-parameter-groups `suivant renvoie une liste de groupes de paramètres.

```
aws memorydb describe-parameter-groups
```

Sortie :

```
{
    "ParameterGroups": [
        {
            "Name": "default.memorydb-redis6",
            "Family": "memorydb_redis6",
            "Description": "Default parameter group for memorydb_redis6",
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/
default.memorydb-redis6"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration des paramètres du moteur à l'aide de groupes de paramètres dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeParameterGroupsà la section Référence des AWS CLI commandes.

`describe-parameters`

L'exemple de code suivant montre comment utiliser`describe-parameters`.

AWS CLI

Pour renvoyer une liste de paramètres

La commande describe-parameters` suivante renvoie une liste de paramètres.

```
aws memorydb describe-parameters
```

Sortie :

```
{
    "Parameters": [
        {
            "Name": "acllog-max-len",
            "Value": "128",
            "Description": "The maximum length of the ACL Log",
            "DataType": "integer",
            "AllowedValues": "1-10000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "activedefrag",
            "Value": "no",
            "Description": "Enabled active memory defragmentation",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-cycle-max",
            "Value": "75",
            "Description": "Maximal effort for defrag in CPU percentage",
            "DataType": "integer",
            "AllowedValues": "1-75",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-cycle-min",
            "Value": "5",
            "Description": "Minimal effort for defrag in CPU percentage",
            "DataType": "integer",
            "AllowedValues": "1-75",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-ignore-bytes",
            "Value": "104857600",
            "Description": "Minimum amount of fragmentation waste to start active
 defrag",
```

```
            "DataType": "integer",
            "AllowedValues": "1048576-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-max-scan-fields",
            "Value": "1000",
            "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
            "DataType": "integer",
            "AllowedValues": "1-1000000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-threshold-lower",
            "Value": "10",
            "Description": "Minimum percentage of fragmentation to start active
defrag",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-threshold-upper",
            "Value": "100",
            "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-expire-effort",
            "Value": "1",
            "Description": "The amount of effort that redis uses to expire items in
the active expiration job",
            "DataType": "integer",
            "AllowedValues": "1-10",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "activerehashing",
            "Value": "yes",
            "Description": "Apply rehashing or not",
```

```
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-hard-limit",
            "Value": "0",
            "Description": "Normal client output buffer hard limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-soft-limit",
            "Value": "0",
            "Description": "Normal client output buffer soft limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-soft-seconds",
            "Value": "0",
            "Description": "Normal client output buffer soft limit in seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-hard-limit",
            "Value": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-soft-limit",
            "Value": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
```

```
        {
            "Name": "client-output-buffer-limit-pubsub-soft-seconds",
            "Value": "60",
            "Description": "Pubsub client output buffer soft limit in seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hash-max-ziplist-entries",
            "Value": "512",
            "Description": "The maximum number of hash entries in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hash-max-ziplist-value",
            "Value": "64",
            "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hll-sparse-max-bytes",
            "Value": "3000",
            "Description": "HyperLogLog sparse representation bytes limit",
            "DataType": "integer",
            "AllowedValues": "1-16000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-eviction",
            "Value": "no",
            "Description": "Perform an asynchronous delete on evictions",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-expire",
```

```
            "Value": "no",
            "Description": "Perform an asynchronous delete on expired keys",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-server-del",
            "Value": "no",
            "Description": "Perform an asynchronous delete on key updates",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-user-del",
            "Value": "no",
            "Description": "Specifies whether the default behavior of DEL command
acts the same as UNLINK",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lfu-decay-time",
            "Value": "1",
            "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policyd",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lfu-log-factor",
            "Value": "10",
            "Description": "The log factor for incrementing key counter for LFU
eviction policy",
            "DataType": "integer",
            "AllowedValues": "1-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "list-compress-depth",
            "Value": "0",
```

```
            "Description": "Number of quicklist ziplist nodes from each side of
 the list to exclude from compression. The head and tail of the list are always
 uncompressed for fast push/pop operations",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "maxmemory-policy",
            "Value": "noeviction",
            "Description": "Max memory policy",
            "DataType": "string",
            "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "maxmemory-samples",
            "Value": "3",
            "Description": "Max memory samples",
            "DataType": "integer",
            "AllowedValues": "1-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "notify-keyspace-events",
            "Description": "The keyspace events for Redis to notify Pub/Sub clients
 about. By default all notifications are disabled",
            "DataType": "string",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "set-max-intset-entries",
            "Value": "512",
            "Description": "The limit in the size of the set in order for the
 dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "slowlog-log-slower-than",
            "Value": "10000",
```

```
            "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command",
            "DataType": "integer",
            "AllowedValues": "-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "slowlog-max-len",
            "Value": "128",
            "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "stream-node-max-bytes",
            "Value": "4096",
            "Description": "The maximum size of a single node in a stream in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "stream-node-max-entries",
            "Value": "100",
            "Description": "The maximum number of items a single node in a stream
can contain",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "tcp-keepalive",
            "Value": "300",
            "Description": "If non-zero, send ACKs every given number of seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "timeout",
```

```
            "Value": "0",
            "Description": "Close connection if client is idle for a given number of
 seconds, or never if 0",
            "DataType": "integer",
            "AllowedValues": "0,20-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "tracking-table-max-keys",
            "Value": "1000000",
            "Description": "The maximum number of keys allowed for the tracking
 table for client side caching",
            "DataType": "integer",
            "AllowedValues": "1-100000000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "zset-max-ziplist-entries",
            "Value": "128",
            "Description": "The maximum number of sorted set entries in order for
 the dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "zset-max-ziplist-value",
            "Value": "64",
            "Description": "The threshold of biggest sorted set entries in order for
 the dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration des paramètres du moteur à l'aide de groupes de paramètres dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeParametersà la section Référence des AWS CLI commandes.

`describe-snapshots`

L'exemple de code suivant montre comment utiliser`describe-snapshots`.

AWS CLI

Pour renvoyer une liste d'instantanés

La commande describe-snapshots` suivante renvoie une liste d'instantanés.

```
aws memorydb describe-snapshots
```

Sortie :

```
{
"Snapshots": [
    {
        "Name": "my-cluster-snapshot",
        "Status": "available",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx2:snapshot/my-cluster-
snapshot",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": " ",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-862574fc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
}
```

Pour plus d'informations, voir Snapshot and restore dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeSnapshotsà la section Référence des AWS CLI commandes.

`describe-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-subnet-groups`.

AWS CLI

Pour renvoyer une liste de groupes de sous-réseaux

Le describe-subnet-groups `suivant renvoie une liste de groupes de sous-réseaux.

```
aws memorydb describe-subnet-groups
```

Sortie

```
{
    "SubnetGroups": [
        {
            "Name": "my-sg",
            "Description": "pat-sg",
            "VpcId": "vpc-86xxx4fc",
            "Subnets": [
                {
                    "Identifier": "subnet-faxx84a6",
                    "AvailabilityZone": {
                        "Name": "us-east-1b"
                    }
                },
                {
                    "Identifier": "subnet-56xxf61b",
                    "AvailabilityZone": {
                        "Name": "us-east-1a"
                    }
                }
            ],
            "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:subnetgroup/my-sg"
        }
    ]
}
```

Pour plus d'informations, consultez la section Sous-réseaux et groupes de sous-réseaux dans le Guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeSubnetGroupsà la section Référence des AWS CLI commandes.

`describe-users`

L'exemple de code suivant montre comment utiliser`describe-users`.

AWS CLI

Pour renvoyer une liste d'utilisateurs

La commande describe-users` suivante renvoie une liste d'utilisateurs.

```
aws memorydb describe-users
```

Sortie

```
{
    "Users": [
        {
            "Name": "default",
            "Status": "active",
            "AccessString": "on ~* &* +@all",
            "ACLNames": [
                "open-access"
            ],
            "MinimumEngineVersion": "6.0",
            "Authentication": {
                "Type": "no-password"
            },
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/default"
        },
        {
            "Name": "my-user",
            "Status": "active",
            "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
            "ACLNames": [],
            "MinimumEngineVersion": "6.2",
            "Authentication": {
                "Type": "password",
                "PasswordCount": 2
            },
```

```
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
        }
    ]
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous DescribeUsersà la section Référence des AWS CLI commandes.

`failover-shard`

L'exemple de code suivant montre comment utiliser`failover-shard`.

AWS CLI

Pour basculer sur un shard

Le failover-shard` suivant échoue sur un shard.

```
aws memorydb failover-shard \
    --cluster-name my-cluster --shard-name 0001
```

Sortie :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "available",
        "NumberOfShards": 2,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SecurityGroups": [
```

```
            {
                "SecurityGroupId": "sg-0a143xxxx45c9fae",
                "Status": "active"
            }
        ],
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour plus d'informations, voir Minimiser les temps d'arrêt avec MultiAZ dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous FailoverShardà la section Référence des AWS CLI commandes.

`list-allowed-node-type-updates`

L'exemple de code suivant montre comment utiliser`list-allowed-node-type-updates`.

AWS CLI

Pour renvoyer une liste des mises à jour de types de nœuds autorisées

L'option list-allowed-node-type -updates suivante renvoie une liste des mises à jour de type de nœud disponibles.

```
aws memorydb list-allowed-node-type-updates
```

Sortie :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "available",
        "NumberOfShards": 2,
        "ClusterEndpoint": {
```

```
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-0a143xxxx45c9fae",
                "Status": "active"
            }
        ],
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour plus d'informations, consultez la section Scaling dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous ListAllowedNodeTypeUpdatesà la section Référence des AWS CLI commandes.

`list-tags`

L'exemple de code suivant montre comment utiliser`list-tags`.

AWS CLI

Pour renvoyer une liste de balises

La liste suivante renvoie une liste de balises.

```
aws memorydb list-tags \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous ListTagsà la section Référence des AWS CLI commandes.

`reset-parameter-group`

L'exemple de code suivant montre comment utiliser`reset-parameter-group`.

AWS CLI

Pour réinitialiser un groupe de paramètres

Le code suivant reset-parameter-group `réinitialise un groupe de paramètres.

```
aws memorydb reset-parameter-group \
    --parameter-group-name my-parameter-group \
    --all-parameters
```

Sortie :

```
{
    "ParameterGroup": {
        "Name": "my-parameter-group",
        "Family": "memorydb_redis6",
        "Description": "my parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-
group"
    }
```

```
}
```

Pour plus d'informations, consultez la section Configuration des paramètres du moteur à l'aide de groupes de paramètres dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous ResetParameterGroupà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une ressource

Le tag-resource` suivant ajoute un tag à une ressource.

```
aws memorydb tag-resource \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster \
    --tags Key="mykey",Value="myvalue"
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        },
        {
            "Key": "mykey",
            "Value": "myvalue"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour mettre à jour une ACL

La commande update-acl` suivante met à jour une ACL en ajoutant un utilisateur.

```
aws memorydb untag-resource \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \
    --tag-keys mykey
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-cluster`

L'exemple de code suivant montre comment utiliser`update-cluster`.

AWS CLI

Pour mettre à jour un cluster

Le cluster`` de mise à jour suivant`` met à jour le groupe de paramètres d'un cluster vers. my-parameter-group

```
aws memorydb update-cluster \
```

```
    --cluster-name my-cluster \
    --parameter-group-name my-parameter-group
```

Sortie :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "available",
        "NumberOfShards": 2,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "my-parameter-group",
        "ParameterGroupStatus": "in-sync",
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-0a143xxxxxc9fae",
                "Status": "active"
            }
        ],
        "SubnetGroupName": "pat-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "ACLName": "my-acl",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour plus d'informations, consultez la section Modification d'un cluster dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous UpdateClusterà la section Référence des AWS CLI commandes.

`update-parameter-group`

L'exemple de code suivant montre comment utiliser`update-parameter-group`.

AWS CLI

Pour mettre à jour un groupe de paramètres

Le update-parameter-group ``suivant met à jour un groupe de paramètres.

```
aws memorydb update-parameter-group \
    --parameter-group-name my-parameter-group \
    --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

Sortie :

```
{
    "ParameterGroup": {
        "Name": "my-parameter-group",
        "Family": "memorydb_redis6",
        "Description": "my parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
    }
}
```

Pour plus d'informations, consultez la section Modification d'un groupe de paramètres dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous UpdateParameterGroupà la section Référence des AWS CLI commandes.

`update-subnet-group`

L'exemple de code suivant montre comment utiliser`update-subnet-group`.

AWS CLI

Pour mettre à jour un groupe de sous-réseaux

Le code suivant update-subnet-group `met à jour l'ID de sous-réseau d'un groupe de sous-réseaux.

```
aws memorydb update-subnet-group \
    --subnet-group-name my-sg \
    --subnet-ids subnet-01f29d458f3xxxxx
```

Sortie :

```
{
    "SubnetGroup": {
        "Name": "my-sg-1",
        "Description": "my-sg",
        "VpcId": "vpc-09d2cfc01xxxxxxx",
        "Subnets": [
            {
                "Identifier": "subnet-01f29d458fxxxxxx",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"
    }
}
```

Pour plus d'informations, consultez la section Sous-réseaux et groupes de sous-réseaux dans le Guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous UpdateSubnetGroupà la section Référence des AWS CLI commandes.

`update-user`

L'exemple de code suivant montre comment utiliser`update-user`.

AWS CLI

Pour mettre à jour un utilisateur

Ce qui suit `update-user` modifie la chaîne d'accès d'un utilisateur.

```
aws memorydb update-user \
    --user-name my-user \
```

```
    --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"
```

Sortie :

```
{
    "User": {
        "Name": "my-user",
        "Status": "modifying",
        "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
        "ACLNames": [
            "myt-acl"
        ],
        "MinimumEngineVersion": "6.2",
        "Authentication": {
            "Type": "password",
            "PasswordCount": 2
        },
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
    }
}
```

Pour plus d'informations, voir Authentification des utilisateurs à l'aide de listes de contrôle d'accès dans le guide de l'utilisateur de MemoryDB.

- Pour plus de détails sur l'API, reportez-vous UpdateUserà la section Référence des AWS CLI commandes.

Exemples d'Amazon MSK utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon MSK.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-cluster`

L'exemple de code suivant montre comment utiliser`create-cluster`.

AWS CLI

Pour créer un cluster Amazon MSK

L'`create-cluster`exemple suivant crée un cluster MSK nommé `MessagingCluster` avec trois nœuds de courtier. Un fichier JSON nommé `brokernodegroupinfo.json` indique les trois sous-réseaux sur lesquels vous souhaitez qu'Amazon MSK distribue les nœuds du broker. Cet exemple ne spécifie pas le niveau de surveillance, c'est donc le cluster qui obtient le `DEFAULT` niveau.

```
aws kafka create-cluster \
    --cluster-name "MessagingCluster" \
    --broker-node-group-info file://brokernodegroupinfo.json \
    --kafka-version "2.2.1" \
    --number-of-broker-nodes 3
```

Contenu de `brokernodegroupinfo.json` :

```
{
    "InstanceType": "kafka.m5.xlarge",
    "BrokerAZDistribution": "DEFAULT",
    "ClientSubnets": [
        "subnet-0123456789111abcd",
        "subnet-0123456789222abcd",
        "subnet-0123456789333abcd"
    ]
}
```

Sortie :

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterName": "MessagingCluster",
    "State": "CREATING"
}
```

Pour plus d'informations, consultez Créer un cluster Amazon MSK dans Amazon Managed Streaming for Apache Kafka Kafka.

- Pour plus de détails sur l'API, reportez-vous CreateClusterà la section Référence des AWS CLI commandes.

`create-configuration`

L'exemple de code suivant montre comment utiliser`create-configuration`.

AWS CLI

Pour créer une configuration Amazon MSK personnalisée

L'`create-configuration`exemple suivant crée une configuration MSK personnalisée avec les propriétés du serveur spécifiées dans le fichier d'entrée.

```
aws kafka create-configuration \
    --name "CustomConfiguration" \
    --description "Topic autocreation enabled; Apache ZooKeeper timeout 2000 ms; Log
 rolling 604800000 ms." \
    --kafka-versions "2.2.1" \
    --server-properties file://configuration.txt
```

Contenu de `configuration.txt` :

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

Cette commande ne produit aucun résultat. Sortie :

```
{
    "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "CreationTime": "2019-10-09T15:26:05.548Z",
    "LatestRevision":
        {
            "CreationTime": "2019-10-09T15:26:05.548Z",
            "Description": "Topic autocreation enabled; Apache ZooKeeper timeout
 2000 ms; Log rolling 604800000 ms.",
            "Revision": 1
```

```
        },
    "Name": "CustomConfiguration"
}
```

Pour plus d'informations, consultez Amazon MSK Configuration Operations dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

- Pour plus de détails sur l'API, reportez-vous CreateConfigurationà la section Référence des AWS CLI commandes.

`describe-cluster`

L'exemple de code suivant montre comment utiliser`describe-cluster`.

AWS CLI

Pour décrire un cluster

L'`describe-cluster`exemple suivant décrit un cluster Amazon MSK.

```
aws kafka describe-cluster \
    --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

Sortie :

```
{
    "ClusterInfo": {
        "BrokerNodeGroupInfo": {
            "BrokerAZDistribution": "DEFAULT",
            "ClientSubnets": [
                "subnet-cbfff283",
                "subnet-6746046b"
            ],
            "InstanceType": "kafka.m5.large",
            "SecurityGroups": [
                "sg-f839b688"
            ],
            "StorageInfo": {
                "EbsStorageInfo": {
                    "VolumeSize": 100
                }
```

```
            }
        },
        "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
        "ClusterName": "demo-cluster-1",
        "CreationTime": "2020-07-09T02:31:36.223000+00:00",
        "CurrentBrokerSoftwareInfo": {
            "KafkaVersion": "2.2.1"
        },
        "CurrentVersion": "K3AEGXETSR30VB",
        "EncryptionInfo": {
            "EncryptionAtRest": {
                "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
            },
            "EncryptionInTransit": {
                "ClientBroker": "TLS_PLAINTEXT",
                "InCluster": true
            }
        },
        "EnhancedMonitoring": "DEFAULT",
        "OpenMonitoring": {
            "Prometheus": {
                "JmxExporter": {
                    "EnabledInBroker": false
                },
                "NodeExporter": {
                    "EnabledInBroker": false
                }
            }
        },
        "NumberOfBrokerNodes": 2,
        "State": "ACTIVE",
        "Tags": {},
        "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
    }
}
```

Pour plus d'informations, consultez la section Listing Amazon MSK Clusters dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

- Pour plus de détails sur l'API, reportez-vous DescribeClusterà la section Référence des AWS CLI commandes.

`get-bootstrap-brokers`

L'exemple de code suivant montre comment utiliser`get-bootstrap-brokers`.

AWS CLI

Pour obtenir des courtiers Bootstrap

L'`get-bootstrap-brokers`exemple suivant récupère les informations du broker bootstrap pour un cluster Amazon MSK.

```
aws kafka get-bootstrap-brokers \
    --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

Sortie :

```
{
    "BootstrapBrokerString": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-east-1.amazonaws.com:9092,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-east-1.amazonaws.com:9092",
    "BootstrapBrokerStringTls": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-east-1.amazonaws.com:9094,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-east-1.amazonaws.com:9094"
}
```

Pour plus d'informations, consultez Getting the Bootstrap Brokers dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

- Pour plus de détails sur l'API, reportez-vous GetBootstrapBrokersà la section Référence des AWS CLI commandes.

`list-clusters`

L'exemple de code suivant montre comment utiliser`list-clusters`.

AWS CLI

Pour répertorier les clusters disponibles

L'`list-clusters`exemple suivant répertorie les clusters Amazon MSK de votre AWS compte.

```
aws kafka list-clusters
```

Sortie :

```
{
    "ClusterInfoList": [
        {
            "BrokerNodeGroupInfo": {
                "BrokerAZDistribution": "DEFAULT",
                "ClientSubnets": [
                    "subnet-cbfff283",
                    "subnet-6746046b"
                ],
                "InstanceType": "kafka.m5.large",
                "SecurityGroups": [
                    "sg-f839b688"
                ],
                "StorageInfo": {
                    "EbsStorageInfo": {
                        "VolumeSize": 100
                    }
                }
            },
            "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
            "ClusterName": "demo-cluster-1",
            "CreationTime": "2020-07-09T02:31:36.223000+00:00",
            "CurrentBrokerSoftwareInfo": {
                "KafkaVersion": "2.2.1"
            },
            "CurrentVersion": "K3AEGXETSR30VB",
            "EncryptionInfo": {
                "EncryptionAtRest": {
                    "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
                },
                "EncryptionInTransit": {
                    "ClientBroker": "TLS_PLAINTEXT",
                    "InCluster": true
                }
            },
```

```
            "EnhancedMonitoring": "DEFAULT",
            "OpenMonitoring": {
                "Prometheus": {
                    "JmxExporter": {
                        "EnabledInBroker": false
                    },
                    "NodeExporter": {
                        "EnabledInBroker": false
                    }
                }
            },
            "NumberOfBrokerNodes": 2,
            "State": "ACTIVE",
            "Tags": {},
            "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
        }
    ]
}
```

Pour plus d'informations, consultez la section Listing Amazon MSK Clusters dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

- Pour plus de détails sur l'API, reportez-vous ListClustersà la section Référence des AWS CLI commandes.

`update-broker-storage`

L'exemple de code suivant montre comment utiliser`update-broker-storage`.

AWS CLI

Pour mettre à jour le stockage EBS pour les courtiers

L'`update-broker-storage`exemple suivant met à jour la quantité de stockage EBS pour tous les courtiers du cluster. Amazon MSK définit la quantité de stockage cible pour chaque courtier selon le montant spécifié dans l'exemple. Vous pouvez obtenir la version actuelle du cluster en décrivant le cluster ou en listant tous les clusters.

```
aws kafka update-broker-storage \
```

```
    --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \
    --current-version "K21V3IB1VIZYYH" \
    --target-broker-ebs-volume-info "KafkaBrokerNodeId=ALL,VolumeSizeGB=1100"
```

La sortie renvoie un ARN pour cette `update-broker-storage` opération. Pour déterminer si cette opération est terminée, utilisez la `describe-cluster-operation` commande avec cet ARN en entrée.

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-
operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-
bcde-33333EXAMPLE"
}
```

Pour plus d'informations, consultez la section Mettre à jour le stockage EBS pour les courtiers dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

- Pour plus de détails sur l'API, reportez-vous UpdateBrokerStorageà la section Référence des AWS CLI commandes.

`update-cluster-configuration`

L'exemple de code suivant montre comment utiliser`update-cluster-configuration`.

AWS CLI

Pour mettre à jour la configuration d'un cluster Amazon MSK

L'`update-cluster-configuration`exemple suivant met à jour la configuration du cluster MSK existant spécifié. Il utilise une configuration MSK personnalisée.

```
aws kafka update-cluster-configuration \
    --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \
    --configuration-info file://configuration-info.json \
    --current-version "K21V3IB1VIZYYH"
```

Contenu de `configuration-info.json` :

```
{
    "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "Revision": 1
}
```

La sortie renvoie un ARN pour cette `update-cluster-configuration` opération. Pour déterminer si cette opération est terminée, utilisez la `describe-cluster-operation` commande avec cet ARN en entrée.

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-
operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-
bcde-33333EXAMPLE"
}
```

Pour plus d'informations, consultez Mettre à jour la configuration d'un cluster Amazon MSK dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

- Pour plus de détails sur l'API, reportez-vous UpdateClusterConfigurationà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Network Flow Monitor AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Network Flow Monitor.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-monitor`

L'exemple de code suivant montre comment utiliser`create-monitor`.

AWS CLI

Pour créer un moniteur

L'`create-monitor`exemple suivant crée un moniteur nommé demo dans le compte spécifié.

```
aws networkflowmonitor create-monitor \
    --monitor-name demo \
    --local-resources type="AWS::EC2::VPC",identifier="arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-03ea55eeda25adbb0"  \
    --scope-arn arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
e21cda79-30a0-4c12-9299-d8629d76d8cf
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/demo",
    "monitorName": "demo",
    "monitorStatus": "ACTIVE",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Créer un moniteur dans Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateMonitorà la section Référence des AWS CLI commandes.

`create-scope`

L'exemple de code suivant montre comment utiliser`create-scope`.

AWS CLI

Pour créer un scope

L'`create-scope`exemple suivant crée une étendue qui inclut un ensemble de ressources pour lesquelles Network Flow Monitor générera des métriques de trafic réseau.

```
aws networkflowmonitor create-scope \
    --targets '[{"targetIdentifier":{"targetId":
{"accountId":"123456789012"},"targetType":"ACCOUNT"},"region":"us-east-1"}]'
```

Sortie :

```
{
    "scopeId": "97626f8d-8a21-4b5d-813a-1a0962dd4615",
    "status": "IN_PROGRESS",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateScopeà la section Référence des AWS CLI commandes.

`delete-monitor`

L'exemple de code suivant montre comment utiliser`delete-monitor`.

AWS CLI

Pour supprimer un moniteur

L'`delete-monitor`exemple suivant supprime un moniteur nommé `demo` dans le compte spécifié.

```
aws networkflowmonitor delete-monitor \
    --monitor-name demo
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un moniteur dans Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteMonitorà la section Référence des AWS CLI commandes.

`delete-scope`

L'exemple de code suivant montre comment utiliser`delete-scope`.

AWS CLI

Pour supprimer une portée

L'`delete-scope`exemple suivant supprime une étendue spécifiée.

```
aws networkflowmonitor delete-scope \
    --scope-id fdc20616-6bb4-4242-a24e-a748e65ca7ac
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteScopeà la section Référence des AWS CLI commandes.

`get-monitor`

L'exemple de code suivant montre comment utiliser`get-monitor`.

AWS CLI

Pour récupérer des informations sur un moniteur

L'`get-monitor`exemple suivant affiche des informations sur le moniteur nommé `demo` dans le compte spécifié.

```
aws networkflowmonitor get-monitor \
    --monitor-name Demo
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo",
    "monitorName": "Demo",
    "monitorStatus": "ACTIVE",
    "localResources": [
```

```
        {
            "type": "AWS::EC2::VPC",
            "identifier": "arn:aws:ec2:us-east-1:123456789012:vpc/
vpc-03ea55eeda25adbb0"
        }
    ],
    "remoteResources": [],
    "createdAt": "2024-12-09T12:21:51.616000-06:00",
    "modifiedAt": "2024-12-09T12:21:55.412000-06:00",
    "tags": {}
}
```

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetMonitorà la section Référence des AWS CLI commandes.

`get-query-results-workload-insights-top-contributors-data`

L'exemple de code suivant montre comment utiliser`get-query-results-workload-insights-top-contributors-data`.

AWS CLI

Pour récupérer les données des principaux contributeurs sur les informations relatives à la charge de travail

L'`get-query-results-workload-insights-top-contributors-data`exemple suivant renvoie les données pour la requête spécifiée.

```
aws networkflowmonitor get-query-results-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id cc4f4ab3-3103-33b8-80ff-d6597a0c6cea
```

Sortie :

```
{
    "datapoints": [
        {
            "timestamps": [
                "2024-12-09T19:00:00+00:00",
```

```
                "2024-12-09T19:05:00+00:00",
                "2024-12-09T19:10:00+00:00"
            ],
            "values": [
                259943.0,
                194856.0,
                216432.0
            ],
            "label": "use1-az6"
        }
    ],
    "unit": "Bytes"
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetQueryResultsWorkloadInsightsTopContributorsDataà la section Référence des AWS CLI commandes.

`get-query-results-workload-insights-top-contributors`

L'exemple de code suivant montre comment utiliser`get-query-results-workload-insights-top-contributors`.

AWS CLI

Pour obtenir des informations sur les principaux contributeurs en matière de charge de travail

L'`get-query-results-workload-insights-top-contributors`exemple suivant renvoie les données pour la requête spécifiée.

```
aws networkflowmonitor get-query-results-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 1fc423d3-b144-37a6-80e6-e2c7d26eea0c
```

Sortie :

```
{
    "topContributors": [
        {
```

```
            "accountId": "123456789012",
            "localSubnetId": "subnet-0a5b30fb95dca2c14",
            "localAz": "use1-az6",
            "localVpcId": "vpc-03ea55eeda25adbb0",
            "localRegion": "us-east-1",
            "remoteIdentifier": "",
            "value": 908443,
            "localSubnetArn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-0a5b30fb95dca2c14",
            "localVpcArn": "arn:aws:ec2:us-east-1:123456789012:vpc/
vpc-03ea55eeda25adbb0"
        }
    ]
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetQueryResultsWorkloadInsightsTopContributorsà la section Référence des AWS CLI commandes.

`get-query-status-monitor-top-contributors`

L'exemple de code suivant montre comment utiliser`get-query-status-monitor-top-contributors`.

AWS CLI

Pour récupérer le statut de la requête

L'`get-query-status-monitor-top-contributors`exemple suivant montre l'état actuel de la requête dans le compte spécifié.

```
aws networkflowmonitor get-query-status-monitor-top-contributors \
    --monitor-name Demo \
    --query-id 5398eabd-bc40-3f5f-aba3-bcb639d3c7ca
```

Sortie :

```
{
    "status": "SUCCEEDED"
```

```
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetQueryStatusMonitorTopContributorsà la section Référence des AWS CLI commandes.

`get-query-status-workload-insights-top-contributors-data`

L'exemple de code suivant montre comment utiliser`get-query-status-workload-insights-top-contributors-data`.

AWS CLI

Pour récupérer le statut de la requête

L'`get-query-status-workload-insights-top-contributors-data`exemple suivant montre l'état actuel de la requête dans le compte spécifié.

```
aws networkflowmonitor get-query-status-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 4333754d-8ae1-3f29-b6b7-c36db2e7f8ac
```

Sortie :

```
{
    "status": "SUCCEEDED"
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetQueryStatusWorkloadInsightsTopContributorsDataà la section Référence des AWS CLI commandes.

`get-query-status-workload-insights-top-contributors`

L'exemple de code suivant montre comment utiliser`get-query-status-workload-insights-top-contributors`.

AWS CLI

Pour récupérer le statut de la requête

L'`get-query-status-workload-insights-top-contributors`exemple suivant montre l'état actuel de la requête dans le compte spécifié.

```
aws networkflowmonitor get-query-status-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id f2a87c70-3e5a-362e-8beb-4747d13d8419
```

Sortie :

```
{
    "status": "SUCCEEDED"
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetQueryStatusWorkloadInsightsTopContributorsà la section Référence des AWS CLI commandes.

`get-scope`

L'exemple de code suivant montre comment utiliser`get-scope`.

AWS CLI

Pour récupérer des informations sur un scope

L'`get-scope`exemple suivant affiche des informations sur une étendue, telles que le statut, les balises, le nom et les détails de la cible.

```
aws networkflowmonitor get-scope \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf
```

Sortie :

```
{
    "scopeId": "e21cda79-30a0-4c12-9299-d8629d76d8cf",
```

```
    "status": "SUCCEEDED",
    "scopeArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
e21cda79-30a0-4c12-9299-d8629d76d8cf",
    "targets": [
        {
            "targetIdentifier": {
                "targetId": {
                    "accountId": "123456789012"
                },
                "targetType": "ACCOUNT"
            },
            "region": "us-east-1"
        }
    ],
    "tags": {}
}
```

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetScopeà la section Référence des AWS CLI commandes.

`list-monitors`

L'exemple de code suivant montre comment utiliser`list-monitors`.

AWS CLI

Pour récupérer une liste de moniteurs

L'`list-monitors`exemple suivant renvoie tous les moniteurs du compte spécifié.

```
aws networkflowmonitor list-monitors
```

Sortie :

```
{
    "monitors": [
        {
            "monitorArn": "arn:aws:networkflowmonitor:us-
east-1:123456789012:monitor/Demo",
```

```
            "monitorName": "Demo",
            "monitorStatus": "ACTIVE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListMonitorsà la section Référence des AWS CLI commandes.

`list-scopes`

L'exemple de code suivant montre comment utiliser`list-scopes`.

AWS CLI

Pour récupérer une liste de scopes

L'`list-scopes`exemple suivant répertorie toutes les étendues du compte spécifié.

```
aws networkflowmonitor list-scopes
```

Sortie :

```
{
    "scopes": [
        {
            "scopeId": "fdc20616-6bb4-4242-a24e-a748e65ca7ac",
            "status": "SUCCEEDED",
            "scopeArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
fdc20616-6bb4-4242-a24e-a748e65ca7ac"
        }
    ]
}
```

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListScopesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags

L'`list-tags-for-resource`exemple suivant renvoie toutes les balises associées à la ressource spécifiée.

```
aws networkflowmonitor list-tags-for-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo
```

Sortie :

```
{
    "tags": {
        "Value": "Production",
        "Key": "stack"
    }
}
```

Pour plus d'informations, consultez la section Marquage de vos CloudWatch ressources Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-query-monitor-top-contributors`

L'exemple de code suivant montre comment utiliser`start-query-monitor-top-contributors`.

AWS CLI

Pour démarrer une requête

L'`start-query-monitor-top-contributors`exemple suivant lance la requête qui renvoie un ID de requête pour récupérer les principaux contributeurs.

```
aws networkflowmonitor start-query-monitor-top-contributors \
    --monitor-name Demo \
```

```
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

Sortie :

```
{
    "queryId": "aecd3a88-0283-35b0-a17d-6e944dc8531d"
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartQueryMonitorTopContributorsà la section Référence des AWS CLI commandes.

`start-query-workload-insights-top-contributors-data`

L'exemple de code suivant montre comment utiliser`start-query-workload-insights-top-contributors-data`.

AWS CLI

Pour démarrer une requête

L'`start-query-workload-insights-top-contributors-data`exemple suivant lance la requête qui renvoie un ID de requête pour récupérer les principaux contributeurs.

```
aws networkflowmonitor start-query-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

Sortie :

```
{
    "queryId": "cc4f4ab3-3103-33b8-80ff-d6597a0c6cea"
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartQueryWorkloadInsightsTopContributorsDataà la section Référence des AWS CLI commandes.

`start-query-workload-insights-top-contributors`

L'exemple de code suivant montre comment utiliser`start-query-workload-insights-top-contributors`.

AWS CLI

Pour démarrer une requête

L'`start-query-workload-insights-top-contributors`exemple suivant lance la requête qui renvoie un ID de requête pour récupérer les principaux contributeurs.

```
aws networkflowmonitor start-query-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

Sortie :

```
{
    "queryId": "1fc423d3-b144-37a6-80e6-e2c7d26eea0c"
}
```

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StartQueryWorkloadInsightsTopContributorsà la section Référence des AWS CLI commandes.

`stop-query-monitor-top-contributors`

L'exemple de code suivant montre comment utiliser`stop-query-monitor-top-contributors`.

AWS CLI

Pour arrêter une requête

L'`stop-query-monitor-top-contributors`exemple suivant arrête la requête dans le compte spécifié.

```
aws networkflowmonitor stop-query-monitor-top-contributors \
    --monitor-name Demo \
    --query-id aecd3a88-0283-35b0-a17d-6e944dc8531d
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopQueryMonitorTopContributorsà la section Référence des AWS CLI commandes.

`stop-query-workload-insights-top-contributors-data`

L'exemple de code suivant montre comment utiliser`stop-query-workload-insights-top-contributors-data`.

AWS CLI

Pour arrêter une requête

L'`stop-query-workload-insights-top-contributors-data`exemple suivant arrête la requête dans le compte spécifié.

```
aws networkflowmonitor stop-query-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id cc4f4ab3-3103-33b8-80ff-d6597a0c6cea
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopQueryWorkloadInsightsTopContributorsDataà la section Référence des AWS CLI commandes.

`stop-query-workload-insights-top-contributors`

L'exemple de code suivant montre comment utiliser`stop-query-workload-insights-top-contributors`.

AWS CLI

Pour arrêter une requête

L'`stop-query-workload-insights-top-contributors`exemple suivant arrête la requête dans le compte spécifié.

```
aws networkflowmonitor stop-query-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 1fc423d3-b144-37a6-80e6-e2c7d26eea0c
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Évaluer les flux réseau à l'aide d'informations sur la charge de travail dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous StopQueryWorkloadInsightsTopContributorsà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter un tag à la ressource spécifiée

L'`tag-resource`exemple suivant ajoute une balise au moniteur dans le compte spécifié.

```
aws networkflowmonitor tag-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo \
    --tags Key=stack,Value=Production
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos CloudWatch ressources Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag de la ressource spécifiée

L'`untag-resource`exemple suivant supprime une balise du moniteur dans le compte spécifié.

```
aws networkflowmonitor untag-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo \
    --tag-keys stack
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage de vos CloudWatch ressources Amazon dans le guide de l' CloudWatch utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-monitor`

L'exemple de code suivant montre comment utiliser`update-monitor`.

AWS CLI

Pour mettre à jour un moniteur existant

L'`update-monitor`exemple suivant met à jour le moniteur nommé `Demo` dans le compte spécifié.

```
aws networkflowmonitor update-monitor \
    --monitor-name Demo \
    --local-resources-to-add type="AWS::EC2::VPC",identifier="arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-048d08dfbec623f94"
```

Sortie :

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo",
    "monitorName": "Demo",
    "monitorStatus": "ACTIVE",
    "tags": {
        "Value": "Production",
        "Key": "stack"
    }
}
```

Pour plus d'informations, consultez la section Composants et fonctionnalités de Network Flow Monitor dans le guide de CloudWatch l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous UpdateMonitorà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Network Manager AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Network Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-customer-gateway`

L'exemple de code suivant montre comment utiliser`associate-customer-gateway`.

AWS CLI

Pour associer une passerelle client

L'`associate-customer-gateway`exemple suivant associe la passerelle client au périphérique `cgw-11223344556677889` dans le réseau mondial spécifié`device-07f6fd08867abc123`.

```
aws networkmanager associate-customer-gateway \
    --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889  \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --region us-west-2
```

Sortie :

```
{
    "CustomerGatewayAssociation": {
        "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "State": "PENDING"
    }
}
```

Pour plus d'informations, consultez la section Customer Gateway Associations dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous AssociateCustomerGatewayà la section Référence des AWS CLI commandes.

`associate-link`

L'exemple de code suivant montre comment utiliser`associate-link`.

AWS CLI

Pour associer un lien

L'`associate-link`exemple suivant associe un lien `link-11112222aaaabbbb1` à un appareil`device-07f6fd08867abc123`. Le lien et le périphérique se trouvent dans le réseau mondial spécifié.

```
aws networkmanager associate-link \
```

```
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

Sortie :

```
{
    "LinkAssociation": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "LinkId": "link-11112222aaaabbbb1",
        "LinkAssociationState": "PENDING"
    }
}
```

Pour plus d'informations, consultez la section Associations de périphériques et de liens dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous AssociateLinkà la section Référence des AWS CLI commandes.

`create-core-network`

L'exemple de code suivant montre comment utiliser`create-core-network`.

AWS CLI

Pour créer un réseau central

L'`create-core-network`exemple suivant crée un réseau central à l'aide d'une description facultative et de balises au sein d'un réseau mondial AWS Cloud WAN.

```
aws networkmanager create-core-network \
    --global-network-id global-network-cdef-EXAMPLE22222 \
    --description "Main headquarters location" \
    --tags Key=Name,Value="New York City office"
```

Sortie :

```
{
```

```
    "CoreNetwork": {
        "GlobalNetworkId": "global-network-cdef-EXAMPLE22222",
        "CoreNetworkId": "core-network-cdef-EXAMPLE33333",
        "CoreNetworkArn": "arn:aws:networkmanager::987654321012:core-network/core-
network-cdef-EXAMPLE33333",
        "Description": "Main headquarters location",
        "CreatedAt": "2022-01-10T19:53:59+00:00",
        "State": "AVAILABLE",
        "Tags": [
            {
                "Key": "Name",
                "Value": "New York City office"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Réseaux mondiaux et principaux dans le Guide de l'utilisateur du AWS Cloud WAN.

- Pour plus de détails sur l'API, reportez-vous CreateCoreNetworkà la section Référence des AWS CLI commandes.

`create-device`

L'exemple de code suivant montre comment utiliser`create-device`.

AWS CLI

Pour créer un appareil

L'`create-device`exemple suivant crée un appareil dans le réseau mondial spécifié. Les détails de l'appareil incluent une description, le type, le fournisseur, le modèle et le numéro de série.

```
aws networkmanager create-device
    --global-network-id global-network-01231231231231231 \
    --description "New York office device" \
    --type "office device" \
    --vendor "anycompany" \
    --model "abcabc" \
    --serial-number "1234" \
    --region us-west-2
```

Sortie :

```
{
    "Device": {
        "DeviceId": "device-07f6fd08867abc123",
        "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York office device",
        "Type": "office device",
        "Vendor": "anycompany",
        "Model": "abcabc",
        "SerialNumber": "1234",
        "CreatedAt": 1575554005.0,
        "State": "PENDING"
    }
}
```

Pour plus d'informations, consultez la section Working with Devices dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous CreateDeviceà la section Référence des AWS CLI commandes.

`create-global-network`

L'exemple de code suivant montre comment utiliser`create-global-network`.

AWS CLI

Pour créer un réseau mondial

Les `create-global-network` exemples suivants créent un nouveau réseau mondial. L'état initial lors de la création est`PENDING`.

```
aws networkmanager create-global-network
```

Sortie :

```
{
    "GlobalNetwork": {
        "GlobalNetworkId": "global-network-00a77fc0f722dae74",
```

```
        "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/
global-network-00a77fc0f722dae74",
        "CreatedAt": "2022-03-14T20:31:56+00:00",
        "State": "PENDING"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateGlobalNetworkà la section Référence des AWS CLI commandes.

`create-link`

L'exemple de code suivant montre comment utiliser`create-link`.

AWS CLI

Pour créer un lien

L'`create-link`exemple suivant crée un lien dans le réseau mondial spécifié. Le lien inclut une description et des détails sur le type de lien, la bande passante et le fournisseur. L'ID du site indique le site auquel le lien est associé.

```
aws networkmanager create-link \
    --global-network-id global-network-01231231231231231 \
    --description "VPN Link" \
    --type "broadband" \
    --bandwidth UploadSpeed=10,DownloadSpeed=20 \
    --provider "AnyCompany" \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

Sortie :

```
{
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
```

```
        "Bandwidth": {
            "UploadSpeed": 10,
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "PENDING"
    }
}
```

Pour plus d'informations, voir Working with Links dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous CreateLinkà la section Référence des AWS CLI commandes.

`create-site`

L'exemple de code suivant montre comment utiliser`create-site`.

AWS CLI

Pour créer un site

L'`create-site`exemple suivant crée un site dans le réseau mondial spécifié. Les détails du site incluent une description et les informations de localisation.

```
aws networkmanager create-site \
    --global-network-id global-network-01231231231231231 \
    --description  "New York head office" \
    --location Latitude=40.7128,Longitude=-74.0060 \
    --region us-west-2
```

Sortie :

```
{
    "Site": {
        "SiteId": "site-444555aaabbb11223",
        "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York head office",
```

```
        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
        "CreatedAt": 1575554300.0,
        "State": "PENDING"
    }
}
```

Pour plus d'informations, consultez la section Working with Sites dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous CreateSiteà la section Référence des AWS CLI commandes.

`create-vpc-attachment`

L'exemple de code suivant montre comment utiliser`create-vpc-attachment`.

AWS CLI

Pour créer une pièce jointe VPC

L'`create-vpc-attachment`exemple suivant crée une attache VPC IPv6 prise en charge par un réseau central.

```
aws networkmanager create-vpc-attachment \
    --core-network-id core-network-0fab62fe438d94db6 \
    --vpc-arn arn:aws:ec2:us-east-1:987654321012:vpc/vpc-09f37f69e2786eeb8  \
    --subnet-arns arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7
 \
    --Ipv6Support=true
```

Sortie :

```
{
    "VpcAttachment": {
        "Attachment": {
            "CoreNetworkId": "core-network-0fab62fe438d94db6",
            "AttachmentId": "attachment-05e1da6eba87a06e6",
            "OwnerAccountId": "987654321012",
            "AttachmentType": "VPC",
```

```
            "State": "CREATING",
            "EdgeLocation": "us-east-1",
            "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/
vpc-09f37f69e2786eeb8",
            "Tags": [],
            "CreatedAt": "2022-03-10T20:59:14+00:00",
            "UpdatedAt": "2022-03-10T20:59:14+00:00"
        },
        "SubnetArns": [
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7"
        ],
        "Options": {
            "Ipv6Support": true
        }
    }
}
```

Pour plus d'informations, consultez la section Créer une pièce jointe dans le guide de l'utilisateur de Cloud WAN.

- Pour plus de détails sur l'API, reportez-vous CreateVpcAttachmentà la section Référence des AWS CLI commandes.

`delete-attachment`

L'exemple de code suivant montre comment utiliser`delete-attachment`.

AWS CLI

Pour supprimer une pièce jointe

L'`delete-attachment`exemple suivant supprime une pièce jointe Connect.

```
aws networkmanager delete-attachment \
    --attachment-id attachment-01feddaeae26ab68c
```

Sortie :

```
{
    "Attachment": {
        "CoreNetworkId": "core-network-0f4b0a9d5ee7761d1",
        "AttachmentId": "attachment-01feddaeae26ab68c",
```

```
        "OwnerAccountId": "987654321012",
        "AttachmentType": "CONNECT",
        "State": "DELETING",
        "EdgeLocation": "us-east-1",
        "ResourceArn": "arn:aws:networkmanager::987654321012:attachment/
attachment-02c3964448fedf5aa",
        "CreatedAt": "2022-03-15T19:18:41+00:00",
        "UpdatedAt": "2022-03-15T19:28:59+00:00"
    }
}
```

Pour plus d'informations, consultez la section Supprimer les pièces jointes dans le guide de l'utilisateur de Cloud WAN.

- Pour plus de détails sur l'API, reportez-vous DeleteAttachmentà la section Référence des AWS CLI commandes.

`delete-bucket-analytics-configuration`

L'exemple de code suivant montre comment utiliser`delete-bucket-analytics-configuration`.

AWS CLI

Pour supprimer une configuration d'analyse pour un bucket

L'`delete-bucket-analytics-configuration`exemple suivant supprime la configuration d'analyse pour le bucket et l'ID spécifiés.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

`delete-bucket-metrics-configuration`

L'exemple de code suivant montre comment utiliser`delete-bucket-metrics-configuration`.

AWS CLI

Pour supprimer la configuration des métriques d'un bucket

L'`delete-bucket-metrics-configuration`exemple suivant supprime la configuration des métriques pour le bucket et l'ID spécifiés.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

`delete-core-network`

L'exemple de code suivant montre comment utiliser`delete-core-network`.

AWS CLI

Pour supprimer un réseau principal

L'`delete-core-network`exemple suivant supprime un réseau central d'un réseau mondial Cloud WAN.

```
aws networkmanager delete-core-network \
    --core-network-id core-network-0fab62fe438d94db6
```

Sortie :

```
{
    "CoreNetwork": {
        "GlobalNetworkId": "global-network-0d59060f16a73bc41",
        "CoreNetworkId": "core-network-0fab62fe438d94db6",
        "Description": "Main headquarters location",
        "CreatedAt": "2021-12-09T18:31:11+00:00",
        "State": "DELETING",
        "Segments": [
```

```
            {
                "Name": "dev",
                "EdgeLocations": [
                    "us-east-1"
                ],
                "SharedSegments": []
            }
        ],
        "Edges": [
            {
                "EdgeLocation": "us-east-1",
                "Asn": 64512,
                "InsideCidrBlocks": []
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Réseaux principaux du Guide de l'utilisateur du Cloud WAN.

- Pour plus de détails sur l'API, reportez-vous DeleteCoreNetworkà la section Référence des AWS CLI commandes.

`delete-device`

L'exemple de code suivant montre comment utiliser`delete-device`.

AWS CLI

Pour supprimer un appareil

L'`delete-device`exemple suivant supprime le périphérique spécifié du réseau mondial spécifié.

```
aws networkmanager delete-device \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --region us-west-2
```

Sortie :

```
{
```

```
    "Device": {
        "DeviceId": "device-07f6fd08867abc123",
        "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York office device",
        "Type": "office device",
        "Vendor": "anycompany",
        "Model": "abcabc",
        "SerialNumber": "1234",
        "SiteId": "site-444555aaabbb11223",
        "CreatedAt": 1575554005.0,
        "State": "DELETING"
    }
}
```

Pour plus d'informations, consultez la section Working with Devices dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteDeviceà la section Référence des AWS CLI commandes.

`delete-global-network`

L'exemple de code suivant montre comment utiliser`delete-global-network`.

AWS CLI

Pour supprimer un réseau mondial

L'`delete-global-network`exemple suivant supprime un réseau mondial.

```
aws networkmanager delete-global-network \
    --global-network-id global-network-052bedddccb193b6b
```

Sortie :

```
{
    "GlobalNetwork": {
        "GlobalNetworkId": "global-network-052bedddccb193b6b",
        "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/
global-network-052bedddccb193b6b",
```

```
        "CreatedAt": "2021-12-09T18:19:12+00:00",
        "State": "DELETING"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteGlobalNetworkà la section Référence des AWS CLI commandes.

`delete-link`

L'exemple de code suivant montre comment utiliser`delete-link`.

AWS CLI

Pour supprimer un lien

L'`delete-link`exemple suivant supprime le lien spécifié du réseau mondial spécifié.

```
aws networkmanager delete-link \
    --global-network-id global-network-01231231231231231  \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

Sortie :

```
{
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
            "UploadSpeed": 20,
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "DELETING"
```

```
    }
}
```

Pour plus d'informations, voir Working with Links dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteLinkà la section Référence des AWS CLI commandes.

`delete-public-access-block`

L'exemple de code suivant montre comment utiliser`delete-public-access-block`.

AWS CLI

Pour supprimer la configuration de blocage de l'accès public pour un bucket

L'`delete-public-access-block`exemple suivant supprime la configuration de blocage de l'accès public sur le compartiment spécifié.

```
aws s3api delete-public-access-block \
    --bucket amzn-s3-demo-bucket
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePublicAccessBlockà la section Référence des AWS CLI commandes.

`delete-site`

L'exemple de code suivant montre comment utiliser`delete-site`.

AWS CLI

Pour supprimer un site

L'`delete-site`exemple suivant supprime le site spécifié (`site-444555aaabbb11223`) dans le réseau mondial spécifié.

```
aws networkmanager delete-site \
```

```
    --global-network-id global-network-01231231231231231  \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

Sortie :

```
{
    "Site": {
        "SiteId": "site-444555aaabbb11223",
        "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York head office",
        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
        "CreatedAt": 1575554300.0,
        "State": "DELETING"
    }
}
```

Pour plus d'informations, consultez la section Working with Sites dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteSiteà la section Référence des AWS CLI commandes.

`deregister-transit-gateway`

L'exemple de code suivant montre comment utiliser`deregister-transit-gateway`.

AWS CLI

Pour désenregistrer une passerelle de transit d'un réseau mondial

L'`deregister-transit-gateway`exemple suivant désenregistre la passerelle de transit spécifiée du réseau mondial spécifié.

```
aws networkmanager deregister-transit-gateway \
    --global-network-id global-network-01231231231231231 \
```

```
    --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc \
    --region us-west-2
```

Sortie :

```
{
    "TransitGatewayRegistration": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
        "State": {
            "Code": "DELETING"
        }
    }
}
```

Pour plus d'informations, consultez la section sur les inscriptions à Transit Gateway dans le guide du gestionnaire du réseau Transit Gateway.

- Pour plus de détails sur l'API, reportez-vous DeregisterTransitGatewayà la section Référence des AWS CLI commandes.

`describe-global-networks`

L'exemple de code suivant montre comment utiliser`describe-global-networks`.

AWS CLI

Pour décrire vos réseaux mondiaux

L'`describe-global-networks`exemple suivant décrit tous les réseaux mondiaux de votre compte.

```
aws networkmanager describe-global-networks \
    --region us-west-2
```

Sortie :

```
{
    "GlobalNetworks": [
```

```
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-
network/global-network-01231231231231231",
            "Description": "Company 1 global network",
            "CreatedAt": 1575553525.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeGlobalNetworksà la section Référence des AWS CLI commandes.

`disassociate-customer-gateway`

L'exemple de code suivant montre comment utiliser`disassociate-customer-gateway`.

AWS CLI

Pour dissocier une passerelle client

L'`disassociate-customer-gateway`exemple suivant dissocie la passerelle client spécifiée (`cgw-11223344556677889`) du réseau mondial spécifié.

```
aws networkmanager disassociate-customer-gateway \
    --global-network-id global-network-01231231231231231 \
    --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
    --region us-west-2
```

Sortie :

```
{
    "CustomerGatewayAssociation": {
        "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "State": "DELETING"
    }
```

```
}
```

Pour plus d'informations, consultez la section Customer Gateway Associations dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous DisassociateCustomerGatewayà la section Référence des AWS CLI commandes.

`disassociate-link`

L'exemple de code suivant montre comment utiliser`disassociate-link`.

AWS CLI

Pour dissocier un lien

L'`disassociate-link`exemple suivant dissocie le lien spécifié du périphérique `device-07f6fd08867abc123` du réseau mondial spécifié.

```
aws networkmanager disassociate-link \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

Sortie :

```
{
    "LinkAssociation": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "LinkId": "link-11112222aaaabbbb1",
        "LinkAssociationState": "DELETING"
    }
}
```

Pour plus d'informations, consultez la section Associations de périphériques et de liens dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous DisassociateLinkà la section Référence des AWS CLI commandes.

`get-bucket-analytics-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-analytics-configuration`.

AWS CLI

Pour récupérer la configuration d'analyse d'un bucket avec un ID spécifique

L'`get-bucket-analytics-configuration`exemple suivant affiche la configuration d'analyse pour le bucket et l'ID spécifiés.

```
aws s3api get-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

Sortie :

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-metrics-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-metrics-configuration`.

AWS CLI

Pour récupérer la configuration des métriques d'un bucket avec un ID spécifique

L'`get-bucket-metrics-configuration`exemple suivant affiche la configuration des métriques pour le bucket et l'ID spécifiés.

```
aws s3api get-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

Sortie :

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

`get-customer-gateway-associations`

L'exemple de code suivant montre comment utiliser`get-customer-gateway-associations`.

AWS CLI

Pour créer des associations avec votre passerelle client

L'`get-customer-gateway-associations`exemple suivant obtient les associations de passerelle client pour le réseau mondial spécifié.

```
aws networkmanager get-customer-gateway-associations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

Sortie :

```
{
    "CustomerGatewayAssociations": [
        {
            "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-
gateway/cgw-11223344556677889",
            "GlobalNetworkId": "global-network-01231231231231231",
            "DeviceId": "device-07f6fd08867abc123",
            "State": "AVAILABLE"
        }
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous GetCustomerGatewayAssociationsà la section Référence des AWS CLI commandes.

`get-devices`

L'exemple de code suivant montre comment utiliser`get-devices`.

AWS CLI

Pour obtenir vos appareils

L'`get-devices`exemple suivant permet d'obtenir les appareils du réseau mondial spécifié.

```
aws networkmanager get-devices \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

Sortie :

```
{
    "Devices": [
        {
            "DeviceId": "device-07f6fd08867abc123",
            "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
            "GlobalNetworkId": "global-network-01231231231231231",
            "Description": "NY office device",
            "Type": "office device",
            "Vendor": "anycompany",
            "Model": "abcabc",
            "SerialNumber": "1234",
            "CreatedAt": 1575554005.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDevicesà la section Référence des AWS CLI commandes.

`get-link-associations`

L'exemple de code suivant montre comment utiliser`get-link-associations`.

AWS CLI

Pour obtenir vos associations de liens

L'`get-link-associations`exemple suivant obtient les associations de liens dans le réseau mondial spécifié.

```
aws networkmanager get-link-associations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

Sortie :

```
{
    "LinkAssociations": [
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "DeviceId": "device-07f6fd08867abc123",
            "LinkId": "link-11112222aaaabbbb1",
            "LinkAssociationState": "AVAILABLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLinkAssociationsà la section Référence des AWS CLI commandes.

`get-links`

L'exemple de code suivant montre comment utiliser`get-links`.

AWS CLI

Pour obtenir vos liens

L'`get-links`exemple suivant permet d'obtenir les liens du réseau mondial spécifié.

```
aws networkmanager get-links \
```

```
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

Sortie :

```
{
    "Links": [
        {
            "LinkId": "link-11112222aaaabbbb1",
            "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
            "GlobalNetworkId": "global-network-01231231231231231",
            "SiteId": "site-444555aaabbb11223",
            "Description": "VPN Link",
            "Type": "broadband",
            "Bandwidth": {
                "UploadSpeed": 10,
                "DownloadSpeed": 20
            },
            "Provider": "AnyCompany",
            "CreatedAt": 1575555811.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetLinksà la section Référence des AWS CLI commandes.

`get-object-retention`

L'exemple de code suivant montre comment utiliser`get-object-retention`.

AWS CLI

Pour récupérer la configuration de rétention d'un objet

L'`get-object-retention`exemple suivant récupère la configuration de rétention d'objets pour l'objet spécifié.

```
aws s3api get-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
```

```
    --key doc1.rtf
```

Sortie :

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS CLI commandes.

`get-public-access-block`

L'exemple de code suivant montre comment utiliser`get-public-access-block`.

AWS CLI

Pour définir ou modifier la configuration de blocage de l'accès public pour un bucket

L'`get-public-access-block`exemple suivant montre la configuration de blocage de l'accès public pour le compartiment spécifié.

```
aws s3api get-public-access-block --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPublicAccessBlockà la section Référence des AWS CLI commandes.

`get-sites`

L'exemple de code suivant montre comment utiliser`get-sites`.

AWS CLI

Pour obtenir vos sites

L'`get-sites`exemple suivant permet d'obtenir les sites du réseau mondial spécifié.

```
aws networkmanager get-sites \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

Sortie :

```
{
    "Sites": [
        {
            "SiteId": "site-444555aaabbb11223",
            "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
            "GlobalNetworkId": "global-network-01231231231231231",
            "Description": "NY head office",
            "Location": {
                "Latitude": "40.7128",
                "Longitude": "-74.0060"
            },
            "CreatedAt": 1575554528.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetSitesà la section Référence des AWS CLI commandes.

`get-transit-gateway-registrations`

L'exemple de code suivant montre comment utiliser`get-transit-gateway-registrations`.

AWS CLI

Pour obtenir les inscriptions à votre passerelle de transit

L'`get-transit-gateway-registrations`exemple suivant obtient les passerelles de transit enregistrées sur le réseau mondial spécifié.

```
aws networkmanager get-transit-gateway-registrations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

Sortie :

```
{
    "TransitGatewayRegistrations": [
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-
gateway/tgw-123abc05e04123abc",
            "State": {
                "Code": "AVAILABLE"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetTransitGatewayRegistrationsà la section Référence des AWS CLI commandes.

`get-vpc-attachment`

L'exemple de code suivant montre comment utiliser`get-vpc-attachment`.

AWS CLI

Pour obtenir une pièce jointe à un VPC

L'`get-vpc-attachment`exemple suivant renvoie des informations sur un attachement VPC.

```
aws networkmanager get-vpc-attachment \
    --attachment-id  attachment-03b7ea450134787da
```

Sortie :

```
{
    "VpcAttachment": {
        "Attachment": {
            "CoreNetworkId": "core-network-0522de1b226a5d7b3",
            "AttachmentId": "attachment-03b7ea450134787da",
            "OwnerAccountId": "987654321012",
            "AttachmentType": "VPC",
            "State": "CREATING",
            "EdgeLocation": "us-east-1",
            "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "DevVPC"
                }
            ],
            "CreatedAt": "2022-03-11T17:48:58+00:00",
            "UpdatedAt": "2022-03-11T17:48:58+00:00"
        },
        "SubnetArns": [
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-202cde6c",
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-e5022dba",
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-2387ae02",
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-cda9dffc"
        ],
        "Options": {
            "Ipv6Support": false
        }
    }
}
```

Pour plus d'informations, consultez la section Pièces jointes du Guide de l'utilisateur du Cloud WAN.

- Pour plus de détails sur l'API, reportez-vous GetVpcAttachmentà la section Référence des AWS CLI commandes.

`list-bucket-analytics-configurations`

L'exemple de code suivant montre comment utiliser`list-bucket-analytics-configurations`.

AWS CLI

Pour récupérer la liste des configurations d'analyse pour un bucket

Ce qui suit `list-bucket-analytics-configurations` permet de récupérer la liste des configurations d'analyse pour le compartiment spécifié.

```
aws s3api list-bucket-analytics-configurations \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketAnalyticsConfigurationsà la section Référence des AWS CLI commandes.

`list-bucket-metrics-configurations`

L'exemple de code suivant montre comment utiliser`list-bucket-metrics-configurations`.

AWS CLI

Pour récupérer la liste des configurations de métriques pour un bucket

L'`list-bucket-metrics-configurations`exemple suivant extrait une liste de configurations de métriques pour le compartiment spécifié.

```
aws s3api list-bucket-metrics-configurations \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
```

```
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketMetricsConfigurationsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises pour la ressource de périphérique spécifiée (`device-07f6fd08867abc123`).

```
aws networkmanager list-tags-for-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 \
    --region us-west-2
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "Network",
```

```
            "Value": "Northeast"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-bucket-metrics-configuration`

L'exemple de code suivant montre comment utiliser`put-bucket-metrics-configuration`.

AWS CLI

Pour définir une configuration de métriques pour un bucket

L'`put-bucket-metrics-configuration`exemple suivant définit une configuration métrique avec l'ID 123 pour le compartiment spécifié.

```
aws s3api put-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

`put-object-retention`

L'exemple de code suivant montre comment utiliser`put-object-retention`.

AWS CLI

Pour définir une configuration de rétention d'objets pour un objet

L'`put-object-retention`exemple suivant définit une configuration de rétention d'objet pour l'objet spécifié jusqu'au 01/01/2025.

```
aws s3api put-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
```

```
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS CLI commandes.

`put-public-access-block`

L'exemple de code suivant montre comment utiliser`put-public-access-block`.

AWS CLI

Pour définir la configuration de blocage de l'accès public pour un bucket

L'`put-public-access-block`exemple suivant définit une configuration d'accès public par blocs restrictifs pour le compartiment spécifié.

```
aws s3api put-public-access-block \
    --bucket amzn-s3-demo-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutPublicAccessBlockà la section Référence des AWS CLI commandes.

`register-transit-gateway`

L'exemple de code suivant montre comment utiliser`register-transit-gateway`.

AWS CLI

Pour enregistrer une passerelle de transit dans un réseau mondial

L'`register-transit-gateway`exemple suivant enregistre une passerelle de transit `tgw-123abc05e04123abc` dans le réseau mondial spécifié.

```
aws networkmanager register-transit-gateway \
    --global-network-id global-network-01231231231231231 \
```

```
    --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc \
    --region us-west-2
```

Sortie :

```
{
    "TransitGatewayRegistration": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
        "State": {
            "Code": "PENDING"
        }
    }
}
```

Pour plus d'informations, consultez la section sur les inscriptions à Transit Gateway dans le guide du gestionnaire du réseau Transit Gateway.

- Pour plus de détails sur l'API, reportez-vous RegisterTransitGatewayà la section Référence des AWS CLI commandes.

`reject-attachment`

L'exemple de code suivant montre comment utiliser`reject-attachment`.

AWS CLI

Pour rejeter une pièce jointe

L'`reject-attachment`exemple suivant rejette une demande d'attachement VPC.

```
aws networkmanager reject-attachment \
    --attachment-id  attachment-03b7ea450134787da
```

Sortie :

```
{
    "Attachment": {
        "CoreNetworkId": "core-network-0522de1b226a5d7b3",
        "AttachmentId": "attachment-03b7ea450134787da",
```

```
        "OwnerAccountId": "987654321012",
        "AttachmentType": "VPC",
        "State": "AVAILABLE",
        "EdgeLocation": "us-east-1",
        "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",
        "CreatedAt": "2022-03-11T17:48:58+00:00",
        "UpdatedAt": "2022-03-11T17:51:25+00:00"
    }
}
```

Pour plus d'informations, consultez la section Acceptation des pièces jointes dans le guide de l'utilisateur du Cloud WAN.

- Pour plus de détails sur l'API, reportez-vous RejectAttachmentà la section Référence des AWS CLI commandes.

`start-route-analysis`

L'exemple de code suivant montre comment utiliser`start-route-analysis`.

AWS CLI

Pour démarrer l'analyse des itinéraires

L'`start-route-analysis`exemple suivant démarre l'analyse entre une source et une destination, y compris l'option facultative`include-return-path`.

```
aws networkmanager start-route-analysis \
    --global-network-id global-network-00aa0aaa0b0aaa000 \
    --source TransitGatewayAttachmentArn=arn:aws:ec2:us-east-1:503089527312:transit-gateway-attachment/tgw-attach-0d4a2d491bf68c093,IpAddress=10.0.0.0 \
    --destination TransitGatewayAttachmentArn=arn:aws:ec2:us-west-1:503089527312:transit-gateway-attachment/tgw-attach-002577f30bb181742,IpAddress=11.0.0.0 \
    --include-return-path
```

Sortie :

```
{
    "RouteAnalysis": {
        "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000
        "OwnerAccountId": "1111222233333",
```

```
        "RouteAnalysisId": "a1873de1-273c-470c-1a2bc2345678",
        "StartTimestamp": 1695760154.0,
        "Status": "RUNNING",
        "Source": {
            "TransitGatewayAttachmentArn": "arn:aws:ec2:us-
east-1:111122223333:transit-gateway-attachment/tgw-attach-1234567890abcdef0,
            "TransitGatewayArn": "arn:aws:ec2:us-east-1:111122223333:transit-
gateway/tgw-abcdef01234567890",
            "IpAddress": "10.0.0.0"
        },
        "Destination": {
            "TransitGatewayAttachmentArn": "arn:aws:ec2:us-
west-1:555555555555:transit-gateway-attachment/tgw-attach-021345abcdef6789",
            "TransitGatewayArn": "arn:aws:ec2:us-west-1:111122223333:transit-
gateway/tgw-09876543210fedcba0",
            "IpAddress": "11.0.0.0"
        },
        "IncludeReturnPath": true,
        "UseMiddleboxes": false
    }
}
```

Pour plus d'informations, consultez Route Analyzer dans le guide de l'utilisateur de AWS Global Networks for Transit Gateways.

- Pour plus de détails sur l'API, reportez-vous StartRouteAnalysisà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour appliquer des balises à une ressource

L'`tag-resource`exemple suivant applique la balise `Network=Northeast` à l'appareil`device-07f6fd08867abc123`.

```
aws networkmanager tag-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 \
    --tags Key=Network,Value=Northeast \
```

```
    --region us-west-2
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`untag-resource`exemple suivant supprime la balise contenant la clé `Network` de l'appareil`device-07f6fd08867abc123`.

```
aws networkmanager untag-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 ]
    --tag-keys Network \
    --region us-west-2
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-device`

L'exemple de code suivant montre comment utiliser`update-device`.

AWS CLI

Pour mettre à jour un appareil

L'`update-device`exemple suivant met à jour l'appareil `device-07f6fd08867abc123` en spécifiant un ID de site pour l'appareil.

```
aws networkmanager update-device \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
```

```
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

Sortie :

```
{
    "Device": {
        "DeviceId": "device-07f6fd08867abc123",
        "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "NY office device",
        "Type": "Office device",
        "Vendor": "anycompany",
        "Model": "abcabc",
        "SerialNumber": "1234",
        "SiteId": "site-444555aaabbb11223",
        "CreatedAt": 1575554005.0,
        "State": "UPDATING"
    }
}
```

Pour plus d'informations, consultez la section Working with Devices dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateDeviceà la section Référence des AWS CLI commandes.

`update-global-network`

L'exemple de code suivant montre comment utiliser`update-global-network`.

AWS CLI

Pour mettre à jour un réseau mondial

L'`update-global-network`exemple suivant met à jour la description du réseau mondial`global-network-01231231231231231`.

```
aws networkmanager update-global-network \
    --global-network-id global-network-01231231231231231 \
    --description "Head offices" \
```

```
    --region us-west-2
```

Sortie :

```
{
    "GlobalNetwork": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/
global-network-01231231231231231",
        "Description": "Head offices",
        "CreatedAt": 1575553525.0,
        "State": "UPDATING"
    }
}
```

Pour plus d'informations, consultez Global Networks dans le guide du gestionnaire de réseau Transit Gateway.

- Pour plus de détails sur l'API, reportez-vous UpdateGlobalNetworkà la section Référence des AWS CLI commandes.

`update-link`

L'exemple de code suivant montre comment utiliser`update-link`.

AWS CLI

Pour mettre à jour un lien

L'`update-link`exemple suivant met à jour les informations de bande passante pour le lien`link-11112222aaaabbbb1`.

```
aws networkmanager update-link \
    --global-network-id global-network-01231231231231231 \
    --link-id link-11112222aaaabbbb1 \
    --bandwidth  UploadSpeed=20,DownloadSpeed=20 \
    --region us-west-2
```

Sortie :

```
{
    "Link": {
```

```
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
            "UploadSpeed": 20,
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "UPDATING"
    }
}
```

Pour plus d'informations, voir Working with Links dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateLinkà la section Référence des AWS CLI commandes.

`update-site`

L'exemple de code suivant montre comment utiliser`update-site`.

AWS CLI

Pour mettre à jour un site

L'`update-site`exemple suivant met à jour la description du site `site-444555aaabbb11223` dans le réseau mondial spécifié.

```
aws networkmanager update-site \
    --global-network-id global-network-01231231231231231 \
    --site-id site-444555aaabbb11223 \
    --description "New York Office site" \
    --region us-west-2
```

Sortie :

```
{
```

```
    "Site": {
        "SiteId": "site-444555aaabbb11223",
        "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York Office site",
        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
        "CreatedAt": 1575554528.0,
        "State": "UPDATING"
    }
}
```

Pour plus d'informations, consultez la section Working with Sites dans le guide du Transit Gateway Network Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateSiteà la section Référence des AWS CLI commandes.

OpenSearch Exemples de services utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du OpenSearch service AWS Command Line Interface with.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-elasticsearch-domain`

L'exemple de code suivant montre comment utiliser`create-elasticsearch-domain`.

AWS CLI

Pour créer un domaine Amazon Elasticsearch Service

La `create-elasticsearch-domain` commande suivante crée un nouveau domaine Amazon Elasticsearch Service au sein d'un VPC et restreint l'accès à un seul utilisateur. Amazon ES déduit l'ID VPC à partir du sous-réseau et du groupe de sécurité spécifiés. IDs

```
aws es create-elasticsearch-domain \
    --domain-name vpc-cli-example \
    --elasticsearch-version 6.2 \
    --elasticsearch-cluster-
config InstanceType=m4.large.elasticsearch,InstanceCount=1 \
    --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": {"AWS": "arn:aws:iam::123456789012:root" }, "Action":"es:*",
 "Resource": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*" } ] }' \
    --vpc-options SubnetIds=subnet-1a2a3a4a,SecurityGroupIds=sg-2a3a4a5a
```

Sortie :

```
{
    "DomainStatus": {
        "ElasticsearchClusterConfig": {
            "DedicatedMasterEnabled": false,
            "InstanceCount": 1,
            "ZoneAwarenessEnabled": false,
            "InstanceType": "m4.large.elasticsearch"
        },
        "DomainId": "123456789012/vpc-cli-example",
        "CognitoOptions": {
            "Enabled": false
        },
        "VPCOptions": {
            "SubnetIds": [
                "subnet-1a2a3a4a"
            ],
            "VPCId": "vpc-3a4a5a6a",
            "SecurityGroupIds": [
                "sg-2a3a4a5a"
            ],
            "AvailabilityZones": [
                "us-west-1c"
```

```
            ]
        },
        "Created": true,
        "Deleted": false,
        "EBSOptions": {
            "VolumeSize": 10,
            "VolumeType": "standard",
            "EBSEnabled": true
        },
        "Processing": true,
        "DomainName": "vpc-cli-example",
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        },
        "ElasticsearchVersion": "6.2",
        "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":
\"es:*\",\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*
\"}]}",
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "EncryptionAtRestOptions": {
            "Enabled": false
        },
        "ARN": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example"
    }
}
```

Pour plus d'informations, consultez Creating and Managing Amazon Elasticsearch Service Domains dans le manuel Amazon Elasticsearch Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateElasticsearchDomainà la section Référence des AWS CLI commandes.

`describe-elasticsearch-domain-config`

L'exemple de code suivant montre comment utiliser`describe-elasticsearch-domain-config`.

AWS CLI

Pour obtenir les détails de configuration du domaine

L'`describe-elasticsearch-domain-config`exemple suivant fournit des détails de configuration pour un domaine donné, ainsi que des informations d'état pour chaque composant de domaine individuel.

```
aws es describe-elasticsearch-domain-config \
    --domain-name cli-example
```

Sortie :

```
{
    "DomainConfig": {
        "ElasticsearchVersion": {
            "Options": "7.4",
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "ElasticsearchClusterConfig": {
            "Options": {
                "InstanceType": "c5.large.elasticsearch",
                "InstanceCount": 1,
                "DedicatedMasterEnabled": true,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "c5.large.elasticsearch",
                "DedicatedMasterCount": 3,
                "WarmEnabled": true,
                "WarmType": "ultrawarm1.medium.elasticsearch",
                "WarmCount": 2
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "EBSOptions": {
            "Options": {
```

```
                "EBSEnabled": true,
                "VolumeType": "gp2",
                "VolumeSize": 10
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "AccessPolicies": {
            "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "SnapshotOptions": {
            "Options": {
                "AutomatedSnapshotStartHour": 0
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "VPCOptions": {
            "Options": {},
            "Status": {
                "CreationDate": 1591210426.162,
                "UpdateDate": 1591210426.162,
                "UpdateVersion": 18,
                "State": "Active",
                "PendingDeletion": false
```

```
            }
        },
        "CognitoOptions": {
            "Options": {
                "Enabled": false
            },
            "Status": {
                "CreationDate": 1591210426.163,
                "UpdateDate": 1591210426.163,
                "UpdateVersion": 18,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "EncryptionAtRestOptions": {
            "Options": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "NodeToNodeEncryptionOptions": {
            "Options": {
                "Enabled": true
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "AdvancedOptions": {
            "Options": {
                "rest.action.multi.allow_explicit_index": "true"
            },
```

```
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "LogPublishingOptions": {
            "Options": {},
            "Status": {
                "CreationDate": 1591210426.164,
                "UpdateDate": 1591210426.164,
                "UpdateVersion": 18,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "DomainEndpointOptions": {
            "Options": {
                "EnforceHTTPS": true,
                "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "AdvancedSecurityOptions": {
            "Options": {
                "Enabled": true,
                "InternalUserDatabaseEnabled": true
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589827485.577,
                "UpdateVersion": 14,
                "State": "Active",
                "PendingDeletion": false
            }
        }
```

```
    }
}
```

Pour plus d'informations, consultez Creating and Managing Amazon Elasticsearch Service Domains dans le manuel Amazon Elasticsearch Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeElasticsearchDomainConfigà la section Référence des AWS CLI commandes.

`describe-elasticsearch-domain`

L'exemple de code suivant montre comment utiliser`describe-elasticsearch-domain`.

AWS CLI

Pour obtenir des informations sur un seul domaine

L'`describe-elasticsearch-domain`exemple suivant fournit les détails de configuration pour un domaine donné.

```
aws es describe-elasticsearch-domain \
    --domain-name cli-example
```

Sortie :

```
{
    "DomainStatus": {
        "DomainId": "123456789012/cli-example",
        "DomainName": "cli-example",
        "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example",
        "Created": true,
        "Deleted": false,
        "Endpoint": "search-cli-example-1a2a3a4a5a6a7a8a9a0a.us-
east-1.es.amazonaws.com",
        "Processing": false,
        "UpgradeProcessing": false,
        "ElasticsearchVersion": "7.4",
        "ElasticsearchClusterConfig": {
            "InstanceType": "c5.large.elasticsearch",
            "InstanceCount": 1,
            "DedicatedMasterEnabled": true,
            "ZoneAwarenessEnabled": false,
            "DedicatedMasterType": "c5.large.elasticsearch",
```

```
            "DedicatedMasterCount": 3,
            "WarmEnabled": true,
            "WarmType": "ultrawarm1.medium.elasticsearch",
            "WarmCount": 2
        },
        "EBSOptions": {
            "EBSEnabled": true,
            "VolumeType": "gp2",
            "VolumeSize": 10
        },
        "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        },
        "CognitoOptions": {
            "Enabled": false
        },
        "EncryptionAtRestOptions": {
            "Enabled": true,
            "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
        },
        "NodeToNodeEncryptionOptions": {
            "Enabled": true
        },
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "ServiceSoftwareOptions": {
            "CurrentVersion": "R20200522",
            "NewVersion": "",
            "UpdateAvailable": false,
            "Cancellable": false,
            "UpdateStatus": "COMPLETED",
            "Description": "There is no software update available for this domain.",
            "AutomatedUpdateDate": 0.0
        },
        "DomainEndpointOptions": {
            "EnforceHTTPS": true,
            "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
        },
        "AdvancedSecurityOptions": {
```

```
            "Enabled": true,
            "InternalUserDatabaseEnabled": true
        }
    }
}
```

Pour plus d'informations, consultez Creating and Managing Amazon Elasticsearch Service Domains dans le manuel Amazon Elasticsearch Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeElasticsearchDomainà la section Référence des AWS CLI commandes.

`describe-elasticsearch-domains`

L'exemple de code suivant montre comment utiliser`describe-elasticsearch-domains`.

AWS CLI

Pour obtenir des informations sur un ou plusieurs domaines

L'`describe-elasticsearch-domains`exemple suivant fournit des détails de configuration pour un ou plusieurs domaines.

```
aws es describe-elasticsearch-domains \
    --domain-names cli-example-1 cli-example-2
```

Sortie :

```
{
    "DomainStatusList": [{
            "DomainId": "123456789012/cli-example-1",
            "DomainName": "cli-example-1",
            "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-1",
            "Created": true,
            "Deleted": false,
            "Endpoint": "search-cli-example-1-1a2a3a4a5a6a7a8a9a0a.us-
east-1.es.amazonaws.com",
            "Processing": false,
            "UpgradeProcessing": false,
            "ElasticsearchVersion": "7.4",
            "ElasticsearchClusterConfig": {
                "InstanceType": "c5.large.elasticsearch",
                "InstanceCount": 1,
```

```
                 "DedicatedMasterEnabled": true,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "c5.large.elasticsearch",
                "DedicatedMasterCount": 3,
                "WarmEnabled": true,
                "WarmType": "ultrawarm1.medium.elasticsearch",
                "WarmCount": 2
            },
            "EBSOptions": {
                "EBSEnabled": true,
                "VolumeType": "gp2",
                "VolumeSize": 10
            },
            "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example-1/*\"}]}",
            "SnapshotOptions": {
                "AutomatedSnapshotStartHour": 0
            },
            "CognitoOptions": {
                "Enabled": false
            },
            "EncryptionAtRestOptions": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
            },
            "NodeToNodeEncryptionOptions": {
                "Enabled": true
            },
            "AdvancedOptions": {
                "rest.action.multi.allow_explicit_index": "true"
            },
            "ServiceSoftwareOptions": {
                "CurrentVersion": "R20200522",
                "NewVersion": "",
                "UpdateAvailable": false,
                "Cancellable": false,
                "UpdateStatus": "COMPLETED",
                "Description": "There is no software update available for this
 domain.",
                "AutomatedUpdateDate": 0.0
            },
            "DomainEndpointOptions": {
```

```
                "EnforceHTTPS": true,
                "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
            },
            "AdvancedSecurityOptions": {
                "Enabled": true,
                "InternalUserDatabaseEnabled": true
            }
        },
        {
            "DomainId": "123456789012/cli-example-2",
            "DomainName": "cli-example-2",
            "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-2",
            "Created": true,
            "Deleted": false,
            "Processing": true,
            "UpgradeProcessing": false,
            "ElasticsearchVersion": "7.4",
            "ElasticsearchClusterConfig": {
                "InstanceType": "r5.large.elasticsearch",
                "InstanceCount": 1,
                "DedicatedMasterEnabled": false,
                "ZoneAwarenessEnabled": false,
                "WarmEnabled": false
            },
            "EBSOptions": {
                "EBSEnabled": true,
                "VolumeType": "gp2",
                "VolumeSize": 10
            },
            "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Deny\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example-2/*\"}]}",
            "SnapshotOptions": {
                "AutomatedSnapshotStartHour": 0
            },
            "CognitoOptions": {
                "Enabled": false
            },
            "EncryptionAtRestOptions": {
                "Enabled": false
            },
            "NodeToNodeEncryptionOptions": {
                "Enabled": false
            },
```

```
            "AdvancedOptions": {
                "rest.action.multi.allow_explicit_index": "true"
            },
            "ServiceSoftwareOptions": {
                "CurrentVersion": "",
                "NewVersion": "",
                "UpdateAvailable": false,
                "Cancellable": false,
                "UpdateStatus": "COMPLETED",
                "Description": "There is no software update available for this
 domain.",
                "AutomatedUpdateDate": 0.0
            },
            "DomainEndpointOptions": {
                "EnforceHTTPS": false,
                "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
            },
            "AdvancedSecurityOptions": {
                "Enabled": false,
                "InternalUserDatabaseEnabled": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez Creating and Managing Amazon Elasticsearch Service Domains dans le manuel Amazon Elasticsearch Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeElasticsearchDomainsà la section Référence des AWS CLI commandes.

`describe-reserved-elasticsearch-instances`

L'exemple de code suivant montre comment utiliser`describe-reserved-elasticsearch-instances`.

AWS CLI

Pour afficher toutes les instances réservées

L'`describe-elasticsearch-domains`exemple suivant fournit un résumé de toutes les instances que vous avez réservées dans une région.

```
aws es describe-reserved-elasticsearch-instances
```

Sortie :

```
{
    "ReservedElasticsearchInstances": [{
        "FixedPrice": 100.0,
        "ReservedElasticsearchInstanceOfferingId":
 "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
        "ReservationName": "my-reservation",
        "PaymentOption": "PARTIAL_UPFRONT",
        "UsagePrice": 0.0,
        "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
        "RecurringCharges": [{
            "RecurringChargeAmount": 0.603,
            "RecurringChargeFrequency": "Hourly"
        }],
        "State": "payment-pending",
        "StartTime": 1522872571.229,
        "ElasticsearchInstanceCount": 3,
        "Duration": 31536000,
        "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
        "CurrencyCode": "USD"
    }]
}
```

Pour plus d'informations, consultez la section Instances réservées dans le manuel Amazon Elasticsearch Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeReservedElasticsearchInstancesà la section Référence des AWS CLI commandes.

`list-domain-names`

L'exemple de code suivant montre comment utiliser`list-domain-names`.

AWS CLI

Pour répertorier tous les domaines

L'`list-domain-names`exemple suivant fournit un bref résumé de tous les domaines de la région.

```
aws es list-domain-names
```

Sortie :

```
{
    "DomainNames": [{
            "DomainName": "cli-example-1"
        },
        {
            "DomainName": "cli-example-2"
        }
    ]
}
```

Pour plus d'informations, consultez Creating and Managing Amazon Elasticsearch Service Domains dans le manuel Amazon Elasticsearch Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListDomainNamesà la section Référence des AWS CLI commandes.

Organisations utilisant des exemples AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Organizations.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-handshake`

L'exemple de code suivant montre comment utiliser`accept-handshake`.

AWS CLI

Pour accepter une poignée de main provenant d'un autre compte

Bill, le propriétaire d'une organisation, a déjà invité le compte de Juan à rejoindre son organisation. L'exemple suivant montre que le compte de Juan accepte la poignée de main et accepte ainsi l'invitation.

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

Le résultat indique ce qui suit :

```
{
        "Handshake": {
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "RequestedTimestamp": 1481656459.257,
                "ExpirationTimestamp": 1482952459.257,
                "Id": "h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "juan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "Resources": [
                        {
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@amazon.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Org Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
```

```
                                        "Value": "ALL"
                                }
                        ],
                        "Type": "ORGANIZATION",
                        "Value": "o-exampleorgid"
                },
                {
                        "Type": "EMAIL",
                        "Value": "juan@example.com"
                }
        ],
        "State": "ACCEPTED"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous AcceptHandshakeà la section Référence des AWS CLI commandes.

`attach-policy`

L'exemple de code suivant montre comment utiliser`attach-policy`.

AWS CLI

Pour associer une politique à une racine, une unité d'organisation ou un compte

Exemple 1

L'exemple suivant montre comment associer une politique de contrôle des services (SCP) à une unité d'organisation :

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

Exemple 2

L'exemple suivant montre comment associer une politique de contrôle des services directement à un compte :

```
aws organizations attach-policy
```

```
    --policy-id p-examplepolicyid111
    --target-id 333333333333
```

- Pour plus de détails sur l'API, reportez-vous AttachPolicyà la section Référence des AWS CLI commandes.

`cancel-handshake`

L'exemple de code suivant montre comment utiliser`cancel-handshake`.

AWS CLI

Pour annuler une poignée de main envoyée depuis un autre compte

Bill avait déjà envoyé une invitation sur le compte de Susan pour rejoindre son organisation. Il change d'avis et décide d'annuler l'invitation avant que Susan ne l'accepte. L'exemple suivant montre l'annulation de Bill :

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

La sortie inclut un objet de poignée de main qui indique que l'état est désormais `CANCELED` le suivant :

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State":"CANCELED",
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "susan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "Resources": [
```

```
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "CONSOLIDATED_BILLING"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "anika@example.com"
                        },
                        {
                                "Type": "NOTES",
                                "Value": "This is a request for Susan's account to
 join Bob's organization."
                        }
                ],
                "RequestedTimestamp": 1.47008383521E9,
                "ExpirationTimestamp": 1.47137983521E9
        }
}
```

- Pour plus de détails sur l'API, reportez-vous CancelHandshakeà la section Référence des AWS CLI commandes.

`create-account`

L'exemple de code suivant montre comment utiliser`create-account`.

AWS CLI

Pour créer un compte membre qui fait automatiquement partie de l'organisation

L'exemple suivant montre comment créer un compte membre dans une organisation. Le compte membre est configuré avec le nom Compte de production et l'adresse e-mail susan@example.com. Organizations crée automatiquement un rôle IAM en utilisant le nom par défaut, OrganizationAccountAccessRole car le paramètre RoleName n'est pas spécifié. En outre, le paramètre qui permet aux utilisateurs ou aux rôles IAM disposant d'autorisations suffisantes d'accéder aux données de facturation du compte est défini sur la valeur par défaut ALLOW car le IamUserAccessToBilling paramètre n'est pas spécifié. Organizations envoie automatiquement à Susan un e-mail « Bienvenue à AWS » :

```
aws organizations create-account --email susan@example.com --account-
name "Production Account"
```

La sortie inclut un objet de demande qui indique que le statut est désormais le suivant `IN_PROGRESS` :

```
{
        "CreateAccountStatus": {
                "State": "IN_PROGRESS",
                "Id": "car-examplecreateaccountrequestid111"
        }
}
```

Vous pouvez ultérieurement demander l'état actuel de la demande en fournissant la valeur de réponse Id à la describe-create-account-status commande comme valeur du create-account-request-id paramètre.

Pour plus d'informations, consultez la section Création d'un AWS compte dans votre organisation dans le Guide de l'utilisateur AWS des Organizations.

- Pour plus de détails sur l'API, reportez-vous CreateAccountà la section Référence des AWS CLI commandes.

`create-organization`

L'exemple de code suivant montre comment utiliser`create-organization`.

AWS CLI

Exemple 1 : pour créer une nouvelle organisation

Bill souhaite créer une organisation à l'aide des informations d'identification du compte 111111111111. L'exemple suivant montre que le compte devient le compte principal de la nouvelle organisation. Comme il ne spécifie aucun ensemble de fonctionnalités, la nouvelle organisation utilise par défaut toutes les fonctionnalités activées et les politiques de contrôle des services sont activées à la racine.

```
aws organizations create-organization
```

La sortie inclut un objet d'organisation contenant des détails sur la nouvelle organisation :

```
{
        "Organization": {
                "AvailablePolicyTypes": [
                        {
                                "Status": "ENABLED",
                                "Type": "SERVICE_CONTROL_POLICY"
                        }
                ],
                "MasterAccountId": "111111111111",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "FeatureSet": "ALL",
                "Id": "o-exampleorgid",
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid"
        }
}
```

Exemple 2 : pour créer une nouvelle organisation avec uniquement les fonctionnalités de facturation consolidée activées

L'exemple suivant crée une organisation qui prend uniquement en charge les fonctionnalités de facturation consolidée :

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

La sortie inclut un objet d'organisation contenant des détails sur la nouvelle organisation :

```
{
```

```
        "Organization": {
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
                "AvailablePolicyTypes": [],
                "Id": "o-exampleorgid",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "MasterAccountId": "111111111111",
                "FeatureSet": "CONSOLIDATED_BILLING"
        }
}
```

Pour plus d'informations, consultez Creating an Organization dans le Guide de l'utilisateur AWS des Organizations.

- Pour plus de détails sur l'API, reportez-vous CreateOrganizationà la section Référence des AWS CLI commandes.

`create-organizational-unit`

L'exemple de code suivant montre comment utiliser`create-organizational-unit`.

AWS CLI

Pour créer une unité d'organisation dans une unité d'organisation racine ou parent

L'exemple suivant montre comment créer une unité d'organisation nommée AccountingOU :

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

La sortie inclut un objet OrganizationalUnit contenant des détails sur la nouvelle unité d'organisation :

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateOrganizationalUnità la section Référence des AWS CLI commandes.

`create-policy`

L'exemple de code suivant montre comment utiliser`create-policy`.

AWS CLI

Exemple 1 : créer une politique avec un fichier source texte pour la politique JSON

L'exemple suivant montre comment créer une politique de contrôle des services (SCP) nomméeAllowAllS3Actions. Le contenu de la politique est extrait d'un fichier sur l'ordinateur local appelépolicy.json.

```
aws organizations create-policy --content file://policy.json --
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
 delegation of all S3 actions"
```

La sortie inclut un objet de stratégie contenant des détails sur la nouvelle stratégie :

```
{
        "Policy": {
                "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*\"]}]}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Allows delegation of all S3 actions",
                        "Name": "AllowAllS3Actions",
                        "Type":"SERVICE_CONTROL_POLICY"
                }
        }
}
```

Exemple 2 : pour créer une politique avec une politique JSON en tant que paramètre

L'exemple suivant montre comment créer le même SCP, cette fois en incorporant le contenu de la politique sous forme de chaîne JSON dans le paramètre. La chaîne doit être évitée par des barres

obliques inverses avant les guillemets doubles pour garantir qu'ils sont traités comme des littéraux dans le paramètre, lui-même entouré de guillemets doubles :

```
aws organizations create-policy --content "{\"Version\":\"2012-10-17\",\"Statement
\":[{\"Effect\":\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*\"]}]}" --
name AllowAllS3Actions --type SERVICE_CONTROL_POLICY --description "Allows
 delegation of all S3 actions"
```

Pour plus d'informations sur la création et l'utilisation de politiques dans votre organisation, consultez la section Managing Organization Policies dans le Guide de l'utilisateur AWS des Organizations.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyà la section Référence des AWS CLI commandes.

`decline-handshake`

L'exemple de code suivant montre comment utiliser`decline-handshake`.

AWS CLI

Pour refuser une poignée de main envoyée depuis un autre compte

L'exemple suivant montre que Susan, une administratrice propriétaire du compte 222222222222, refuse une invitation à rejoindre l'organisation de Bill. L' DeclineHandshake opération renvoie un objet de poignée de main, indiquant que l'état est désormais REFUSÉ :

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

La sortie inclut un objet de poignée de main qui indique le nouvel état de `DECLINED` :

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State": "DECLINED",
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
```

```
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "susan@example.com"
                        },
                        {
                                "Type": "NOTES",
                                "Value": "This is an invitation to Susan's account
 to join the Bill's organization."
                        }
                ],
                "Parties": [
                        {
                                "Type": "EMAIL",
                                "Id": "susan@example.com"
                        },
                        {
                                "Type": "ORGANIZATION",
                                "Id": "o-exampleorgid"
                        }
                ],
                "Action": "INVITE",
                "RequestedTimestamp": 1470684478.687,
                "ExpirationTimestamp": 1471980478.687,
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous DeclineHandshakeà la section Référence des AWS CLI commandes.

`delete-organization`

L'exemple de code suivant montre comment utiliser`delete-organization`.

AWS CLI

Pour supprimer une organisation

L'exemple suivant montre comment supprimer une organisation. Pour effectuer cette opération, vous devez être administrateur du compte principal de l'organisation. L'exemple suppose que vous avez précédemment supprimé tous les comptes et politiques des membres de l'organisation : OUs

```
aws organizations delete-organization
```

- Pour plus de détails sur l'API, reportez-vous DeleteOrganizationà la section Référence des AWS CLI commandes.

`delete-organizational-unit`

L'exemple de code suivant montre comment utiliser`delete-organizational-unit`.

AWS CLI

Pour supprimer une UO

L'exemple suivant montre comment supprimer une unité d'organisation. L'exemple suppose que vous avez précédemment supprimé tous les comptes et autres comptes OUs de l'unité d'organisation :

```
aws organizations delete-organizational-unit --organizational-unit-id ou-
examplerootid111-exampleouid111
```

- Pour plus de détails sur l'API, reportez-vous DeleteOrganizationalUnità la section Référence des AWS CLI commandes.

`delete-policy`

L'exemple de code suivant montre comment utiliser`delete-policy`.

AWS CLI

Pour supprimer une politique

L'exemple suivant montre comment supprimer une politique d'une organisation. L'exemple suppose que vous avez précédemment détaché la politique de toutes les entités :

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`describe-account`

L'exemple de code suivant montre comment utiliser`describe-account`.

AWS CLI

Pour obtenir les informations relatives à un compte

L'exemple suivant vous montre comment demander des informations sur un compte :

```
aws organizations describe-account --account-id 555555555555
```

La sortie montre un objet de compte avec les détails du compte :

```
{
        "Account": {
                "Id": "555555555555",
                "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/555555555555",
                "Name": "Beta account",
                "Email": "anika@example.com",
                "JoinedMethod": "INVITED",
                "JoinedTimeStamp": 1481756563.134,
                "Status": "ACTIVE"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAccountà la section Référence des AWS CLI commandes.

`describe-create-account-status`

L'exemple de code suivant montre comment utiliser`describe-create-account-status`.

AWS CLI

Pour connaître le statut le plus récent d'une demande de création de compte

L'exemple suivant montre comment demander le dernier statut d'une demande précédente de création de compte dans une organisation. Le --request-id spécifié provient de la réponse de l'appel initial à create-account. La demande de création de compte indique par le champ de statut que les Organisations ont terminé avec succès la création du compte.

Commande :

```
aws organizations describe-create-account-status --create-account-request-id car-examplecreateaccountrequestid111
```

Sortie :

```
{
  "CreateAccountStatus": {
    "State": "SUCCEEDED",
    "AccountId": "555555555555",
    "AccountName": "Beta account",
    "RequestedTimestamp": 1470684478.687,
    "CompletedTimestamp": 1470684532.472,
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeCreateAccountStatusà la section Référence des AWS CLI commandes.

`describe-handshake`

L'exemple de code suivant montre comment utiliser`describe-handshake`.

AWS CLI

Pour obtenir des informations sur une poignée de main

L'exemple suivant montre comment demander des informations sur une poignée de main. L'identifiant de poignée de main provient soit de l'appel initial

versInviteAccountToOrganization, soit d'un appel vers ListHandshakesForAccount ou ListHandshakesForOrganization :

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

La sortie inclut un objet de poignée de main contenant tous les détails sur la poignée de main demandée :

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State": "OPEN",
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "anika@example.com"
                        }
                ],
                "Parties": [
                        {
                                "Type": "ORGANIZATION",
                                "Id": "o-exampleorgid"
                        },
                        {
                                "Type": "EMAIL",
                                "Id": "anika@example.com"
                        }
                ],
                "Action": "INVITE",
```

```
                "RequestedTimestamp": 1470158698.046,
                "ExpirationTimestamp": 1471454698.046,
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeHandshakeà la section Référence des AWS CLI commandes.

`describe-organization`

L'exemple de code suivant montre comment utiliser`describe-organization`.

AWS CLI

Pour obtenir des informations sur l'organisation actuelle

L'exemple suivant vous montre comment demander des informations sur une organisation :

```
aws organizations describe-organization
```

La sortie inclut un objet d'organisation contenant les informations relatives à l'organisation :

```
{
        "Organization": {
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "MasterAccountId": "111111111111",
                "Id": "o-exampleorgid",
                "FeatureSet": "ALL",
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
                "AvailablePolicyTypes": [
                        {
                                "Status": "ENABLED",
                                "Type": "SERVICE_CONTROL_POLICY"
                        }
                ]
        }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeOrganizationà la section Référence des AWS CLI commandes.

`describe-organizational-unit`

L'exemple de code suivant montre comment utiliser`describe-organizational-unit`.

AWS CLI

Pour obtenir des informations sur une UO

L'`describe-organizational-unit`exemple suivant demande des informations sur une unité d'organisation.

```
aws organizations describe-organizational-unit \
    --organizational-unit-id ou-examplerootid111-exampleouid111
```

Sortie :

```
{
    "OrganizationalUnit": {
        "Name": "Accounting Group",
        "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
        "Id": "ou-examplerootid111-exampleouid111"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeOrganizationalUnità la section Référence des AWS CLI commandes.

`describe-policy`

L'exemple de code suivant montre comment utiliser`describe-policy`.

AWS CLI

Pour obtenir des informations sur une politique

L'exemple suivant montre comment demander des informations sur une politique :

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

La sortie inclut un objet de stratégie qui contient des détails sur la stratégie :

```
{
        "Policy": {
                "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n
    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n      \"Resource\":
 \"*\"\n    }\n  ]\n}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::111111111111:policy/o-
exampleorgid/service_control_policy/p-examplepolicyid111",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-examplepolicyid111",
                        "AwsManaged": false,
                        "Name": "AllowAllS3Actions",
                        "Description": "Enables admins to delegate S3 permissions"
                }
        }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePolicyà la section Référence des AWS CLI commandes.

`detach-policy`

L'exemple de code suivant montre comment utiliser`detach-policy`.

AWS CLI

Pour détacher une politique d'une racine, d'une unité d'organisation ou d'un compte

L'exemple suivant montre comment détacher une politique d'une unité d'organisation :

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111 --
policy-id p-examplepolicyid111
```

- Pour plus de détails sur l'API, reportez-vous DetachPolicyà la section Référence des AWS CLI commandes.

`disable-policy-type`

L'exemple de code suivant montre comment utiliser`disable-policy-type`.

AWS CLI

Pour désactiver un type de politique dans une racine

L'exemple suivant montre comment désactiver le type de politique de contrôle des services (SCP) dans une racine :

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

Le résultat indique que l'élément de PolicyTypes réponse n'inclut plus SERVICE_CONTROL_POLICY :

```
{
        "Root": {
                "PolicyTypes": [],
                "Name": "Root",
                "Id": "r-examplerootid111",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous DisablePolicyTypeà la section Référence des AWS CLI commandes.

`enable-all-features`

L'exemple de code suivant montre comment utiliser`enable-all-features`.

AWS CLI

Pour activer toutes les fonctionnalités d'une organisation

Cet exemple montre que l'administrateur demande à tous les comptes invités de l'organisation d'approuver l'activation de toutes les fonctionnalités de l'organisation. AWS Organizations envoie un e-mail à l'adresse enregistrée avec chaque compte de membre invité demandant au

propriétaire d'approuver la modification de toutes les fonctionnalités en acceptant la poignée de main envoyée. Une fois que tous les comptes de membres invités ont accepté la poignée de main, l'administrateur de l'organisation peut finaliser la modification de toutes les fonctionnalités, et ceux qui disposent des autorisations appropriées peuvent créer des politiques et les appliquer aux racines et aux comptes : OUs

```
aws organizations enable-all-features
```

Le résultat est un objet de poignée de main qui est envoyé à tous les comptes de membres invités pour approbation :

```
{
        "Handshake": {
                "Action": "ENABLE_ALL_FEATURES",
                "Arn":"arn:aws:organizations::111111111111:handshake/o-exampleorgid/
enable_all_features/h-examplehandshakeid111",
                "ExpirationTimestamp":1.483127868609E9,
                "Id":"h-examplehandshakeid111",
                "Parties": [
                        {
                                "id":"o-exampleorgid",
                                "type":"ORGANIZATION"
                        }
                ],
                "requestedTimestamp":1.481831868609E9,
                "resources": [
                        {
                                "type":"ORGANIZATION",
                                "value":"o-exampleorgid"
                        }
                ],
                "state":"REQUESTED"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous EnableAllFeaturesà la section Référence des AWS CLI commandes.

`enable-policy-type`

L'exemple de code suivant montre comment utiliser`enable-policy-type`.

AWS CLI

Pour activer l'utilisation d'un type de politique dans une racine

L'exemple suivant montre comment activer le type de politique de contrôle des services (SCP) dans une racine :

```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

La sortie montre un objet racine avec un élément de réponse PolicyTypes indiquant qu' SCPs ils sont désormais activés :

```
{
        "Root": {
                "PolicyTypes": [
                        {
                                "Status":"ENABLED",
                                "Type":"SERVICE_CONTROL_POLICY"
                        }
                ],
                "Id": "r-examplerootid111",
                "Name": "Root",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous EnablePolicyTypeà la section Référence des AWS CLI commandes.

`invite-account-to-organization`

L'exemple de code suivant montre comment utiliser`invite-account-to-organization`.

AWS CLI

Pour inviter un compte à rejoindre une organisation

L'exemple suivant montre le compte principal appartenant à bill@example.com invitant le compte détenu par juan@example.com à rejoindre une organisation :

```
aws organizations invite-account-to-organization --target '{"Type": "EMAIL", "Id":
 "juan@example.com"}' --notes "This is a request for Juan's account to join Bill's
 organization."
```

Le résultat inclut une structure de poignée de main qui montre ce qui est envoyé au compte invité :

```
{
        "Handshake": {
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "ExpirationTimestamp": 1482952459.257,
                "Id": "h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "juan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "RequestedTimestamp": 1481656459.257,
                "Resources": [
                        {
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@amazon.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Org Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "FULL"
                                        }
                                ],
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid"
```

```
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "juan@example.com"
                        }
                ],
                "State": "OPEN"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous InviteAccountToOrganizationà la section Référence des AWS CLI commandes.

`leave-organization`

L'exemple de code suivant montre comment utiliser`leave-organization`.

AWS CLI

Pour quitter une organisation en tant que compte membre

L'exemple suivant montre l'administrateur d'un compte membre demandant à quitter l'organisation dont il est actuellement membre :

```
aws organizations leave-organization
```

- Pour plus de détails sur l'API, reportez-vous LeaveOrganizationà la section Référence des AWS CLI commandes.

`list-accounts-for-parent`

L'exemple de code suivant montre comment utiliser`list-accounts-for-parent`.

AWS CLI

Pour récupérer la liste de tous les comptes d'une racine parent ou d'une unité d'organisation spécifiée

L'exemple suivant montre comment demander la liste des comptes d'une unité d'organisation :

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-exampleouid111
```

La sortie inclut une liste d'objets récapitulatifs du compte.

```
{
        "Accounts": [
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-exampleorgid/333333333333",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835795.536,
                        "Id": "333333333333",
                        "Name": "Development Account",
                        "Email": "juan@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-exampleorgid/444444444444",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835812.143,
                        "Id": "444444444444",
                        "Name": "Test Account",
                        "Email": "anika@example.com",
                        "Status": "ACTIVE"
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAccountsForParentà la section Référence des AWS CLI commandes.

`list-accounts`

L'exemple de code suivant montre comment utiliser`list-accounts`.

AWS CLI

Pour récupérer la liste de tous les comptes d'une organisation

L'exemple suivant montre comment demander une liste des comptes d'une organisation :

```
aws organizations list-accounts
```

La sortie inclut une liste d'objets récapitulatifs du compte.

```
{
        "Accounts": [
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481830215.45,
                        "Id": "111111111111",
                        "Name": "Master Account",
                        "Email": "bill@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/222222222222",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835741.044,
                        "Id": "222222222222",
                        "Name": "Production Account",
                        "Email": "alice@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835795.536,
                        "Id": "333333333333",
                        "Name": "Development Account",
                        "Email": "juan@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/444444444444",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835812.143,
                        "Id": "444444444444",
                        "Name": "Test Account",
```

```
                        "Email": "anika@example.com",
                        "Status": "ACTIVE"
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAccountsà la section Référence des AWS CLI commandes.

`list-children`

L'exemple de code suivant montre comment utiliser`list-children`.

AWS CLI

Pour récupérer les comptes enfants et ceux OUs d'une unité d'organisation ou root parent

L'exemple suivant explique comment répertorier la racine ou l'unité d'organisation contenant ce compte 444444444444 :

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-examplerootid111-exampleouid111
```

La sortie montre les deux enfants OUs contenus par le parent :

```
{
        "Children": [
                {
                        "Id": "ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                },
                {
                        "Id":"ou-examplerootid111-exampleouid222",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListChildrenà la section Référence des AWS CLI commandes.

`list-create-account-status`

L'exemple de code suivant montre comment utiliser`list-create-account-status`.

AWS CLI

Exemple 1 : Pour récupérer la liste des demandes de création de compte effectuées dans l'organisation actuelle

L'exemple suivant montre comment demander une liste des demandes de création de compte pour une organisation qui ont été traitées avec succès :

```
aws organizations list-create-account-status --states SUCCEEDED
```

La sortie inclut un ensemble d'objets contenant des informations sur chaque demande.

```
{
        "CreateAccountStatuses": [
                {
                        "AccountId": "444444444444",
                        "AccountName": "Developer Test Account",
                        "CompletedTimeStamp": 1481835812.143,
                        "Id": "car-examplecreateaccountrequestid111",
                        "RequestedTimeStamp": 1481829432.531,
                        "State": "SUCCEEDED"
                }
        ]
}
```

Exemple 2 : Pour récupérer la liste des demandes de création de compte en cours effectuées dans l'organisation actuelle

L'exemple suivant permet d'obtenir la liste des demandes de création de compte en cours pour une organisation :

```
aws organizations list-create-account-status --states IN_PROGRESS
```

La sortie inclut un ensemble d'objets contenant des informations sur chaque demande.

```
{
        "CreateAccountStatuses": [
```

```
                {
                  "State": "IN_PROGRESS",
                  "Id": "car-examplecreateaccountrequestid111",
                  "RequestedTimeStamp": 1481829432.531,
                  "AccountName": "Production Account"
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListCreateAccountStatusà la section Référence des AWS CLI commandes.

`list-handshakes-for-account`

L'exemple de code suivant montre comment utiliser`list-handshakes-for-account`.

AWS CLI

Pour récupérer la liste des poignées de main envoyées à un compte

L'exemple suivant montre comment obtenir une liste de toutes les poignées de main associées au compte des informations d'identification utilisées pour appeler l'opération :

```
aws organizations list-handshakes-for-account
```

La sortie inclut une liste de structures de poignée de main avec des informations sur chaque poignée de main, y compris son état actuel :

```
{
        "Handshake": {
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "ExpirationTimestamp": 1482952459.257,
                "Id": "h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
```

```
                        "Id": "juan@example.com",
                        "Type": "EMAIL"
                }
            ],
            "RequestedTimestamp": 1481656459.257,
            "Resources": [
                {
                    "Resources": [
                        {
                            "Type": "MASTER_EMAIL",
                            "Value": "bill@amazon.com"
                        },
                        {
                            "Type": "MASTER_NAME",
                            "Value": "Org Master Account"
                        },
                        {
                            "Type": "ORGANIZATION_FEATURE_SET",
                            "Value": "FULL"
                        }
                    ],
                    "Type": "ORGANIZATION",
                    "Value": "o-exampleorgid"
                },
                {
                    "Type": "EMAIL",
                    "Value": "juan@example.com"
                }
            ],
            "State": "OPEN"
        }
}
```

- Pour plus de détails sur l'API, reportez-vous ListHandshakesForAccountà la section Référence des AWS CLI commandes.

`list-handshakes-for-organization`

L'exemple de code suivant montre comment utiliser`list-handshakes-for-organization`.

AWS CLI

Pour récupérer la liste des poignées de main associées à une organisation

L'exemple suivant montre comment obtenir une liste des poignées de main associées à l'organisation actuelle :

```
aws organizations list-handshakes-for-organization
```

La sortie montre deux poignées de main. Le premier est une invitation à accéder au compte de Juan et indique l'état OUVERT. La seconde est une invitation à accéder au compte d'Anika et indique que l'état est ACCEPTÉ :

```
{
        "Handshakes": [
                {
                        "Action": "INVITE",
                        "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                        "ExpirationTimestamp": 1482952459.257,
                        "Id": "h-examplehandshakeid111",
                        "Parties": [
                                {
                                        "Id": "o-exampleorgid",
                                        "Type": "ORGANIZATION"
                                },
                                {
                                        "Id": "juan@example.com",
                                        "Type": "EMAIL"
                                }
                        ],
                        "RequestedTimestamp": 1481656459.257,
                        "Resources": [
                                {
                                        "Resources": [
                                                {
                                                        "Type": "MASTER_EMAIL",
                                                        "Value": "bill@amazon.com"
                                                },
                                                {
                                                        "Type": "MASTER_NAME",
                                                        "Value": "Org Master
 Account"
                                                },
                                                {
                                                        "Type":
 "ORGANIZATION_FEATURE_SET",
```

```
                                                        "Value": "FULL"
                                                }
                                        ],
                                        "Type": "ORGANIZATION",
                                        "Value": "o-exampleorgid"
                                },
                                {
                                        "Type": "EMAIL",
                                        "Value": "juan@example.com"
                                },
                                {
                                        "Type":"NOTES",
                                        "Value":"This is an invitation to Juan's
 account to join Bill's organization."
                                }
                        ],
                        "State": "OPEN"
                },
                {
                        "Action": "INVITE",
                        "State":"ACCEPTED",
                        "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                        "ExpirationTimestamp": 1.471797437427E9,
                        "Id": "h-examplehandshakeid222",
                        "Parties": [
                                {
                                        "Id": "o-exampleorgid",
                                        "Type": "ORGANIZATION"
                                },
                                {
                                        "Id": "anika@example.com",
                                        "Type": "EMAIL"
                                }
                        ],
                        "RequestedTimestamp": 1.469205437427E9,
                        "Resources": [
                                {
                                        "Resources": [
                                                {
                                                        "Type":"MASTER_EMAIL",
                                                        "Value":"bill@example.com"
                                                },
                                                {
```

```
                                                "Type":"MASTER_NAME",
                                                "Value":"Master Account"
                                        }
                                ],
                                "Type":"ORGANIZATION",
                                "Value":"o-exampleorgid"
                        },
                        {
                                "Type":"EMAIL",
                                "Value":"anika@example.com"
                        },
                        {
                                "Type":"NOTES",
                                "Value":"This is an invitation to Anika's
 account to join Bill's organization."
                        }
                ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListHandshakesForOrganizationà la section Référence des AWS CLI commandes.

`list-organizational-units-for-parent`

L'exemple de code suivant montre comment utiliser`list-organizational-units-for-parent`.

AWS CLI

Pour récupérer une liste des OUs dans une unité d'organisation parent ou racine

L'exemple suivant vous montre comment obtenir une liste de OUs dans une racine spécifiée :

```
aws organizations list-organizational-units-for-parent --parent-id r-
examplerootid111
```

Le résultat indique que la racine spécifiée en contient deux OUs et indique le détail de chacune d'entre elles :

```
{
        "OrganizationalUnits": [
```

```
                {
                        "Name": "AccountingDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid111"
                },
                {
                        "Name": "ProductionDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid222"
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListOrganizationalUnitsForParentà la section Référence des AWS CLI commandes.

`list-parents`

L'exemple de code suivant montre comment utiliser`list-parents`.

AWS CLI

Pour répertorier le parent OUs ou les racines d'un compte ou d'une unité d'organisation enfant

L'exemple suivant explique comment répertorier l'unité d'organisation racine ou parent qui contient ce compte 444444444444 :

```
aws organizations list-parents --child-id 444444444444
```

La sortie indique que le compte spécifié se trouve dans l'unité d'organisation avec l'ID spécifié :

```
{
  "Parents": [
        {
          "Id": "ou-examplerootid111-exampleouid111",
          "Type": "ORGANIZATIONAL_UNIT"
        }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListParentsà la section Référence des AWS CLI commandes.

`list-policies-for-target`

L'exemple de code suivant montre comment utiliser`list-policies-for-target`.

AWS CLI

Pour récupérer la liste des pièces SCPs jointes directement à un compte

L'exemple suivant montre comment obtenir une liste de toutes les politiques de contrôle des services (SCPs), telles que spécifiées par le paramètre Filter, qui sont directement associées à un compte :

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-
id 444444444444
```

Le résultat inclut une liste de structures politiques avec des informations récapitulatives sur les politiques. La liste n'inclut pas les politiques qui s'appliquent au compte en raison de l'héritage provenant de son emplacement dans une hiérarchie d'unités d'organisation :

```
{
        "Policies": [
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllEC2Actions",
                        "AwsManaged", false,
                        "Id": "p-examplepolicyid222",
                        "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid222",
                        "Description": "Enables account admins to delegate
 permissions for any EC2 actions to users and roles in their accounts."
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPoliciesForTargetà la section Référence des AWS CLI commandes.

`list-policies`

L'exemple de code suivant montre comment utiliser`list-policies`.

AWS CLI

Pour récupérer la liste de toutes les politiques d'une organisation d'un certain type

L'exemple suivant montre comment obtenir une liste de SCPs, comme indiqué par le paramètre de filtre :

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

Le résultat inclut une liste de politiques avec des informations récapitulatives :

```
{
        "Policies": [
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllS3Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid111",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Enables account admins to delegate
 permissions for any S3 actions to users and roles in their accounts."
                },
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllEC2Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid222",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid222",
                        "Description": "Enables account admins to delegate
 permissions for any EC2 actions to users and roles in their accounts."
                },
                {
                        "AwsManaged": true,
                        "Description": "Allows access to every operation",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-FullAWSAccess",
                        "Arn": "arn:aws:organizations::aws:policy/
service_control_policy/p-FullAWSAccess",
                        "Name": "FullAWSAccess"
                }
        ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListPoliciesà la section Référence des AWS CLI commandes.

`list-roots`

L'exemple de code suivant montre comment utiliser`list-roots`.

AWS CLI

Pour récupérer la liste des racines d'une organisation

Cet exemple vous montre comment obtenir la liste des racines d'une organisation :

```
aws organizations list-roots
```

La sortie inclut une liste de structures racines avec des informations récapitulatives :

```
{
        "Roots": [
                {
                        "Name": "Root",
                        "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
                        "Id": "r-examplerootid111",
                        "PolicyTypes": [
                                {
                                        "Status":"ENABLED",
                                        "Type":"SERVICE_CONTROL_POLICY"
                                }
                        ]
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListRootsà la section Référence des AWS CLI commandes.

`list-targets-for-policy`

L'exemple de code suivant montre comment utiliser`list-targets-for-policy`.

AWS CLI

Pour récupérer la liste des racines et OUs des comptes auxquels une politique est attachée

L'exemple suivant montre comment obtenir une liste des racines et OUs des comptes auxquels la politique spécifiée est attachée :

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

La sortie inclut une liste d'objets de pièce jointe avec des informations récapitulatives sur les OUs racines et les comptes auxquels la politique est attachée :

```
{
        "Targets": [
                {
                        "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
                        "Name": "Root",
                        "TargetId":"r-examplerootid111",
                        "Type":"ROOT"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333;",
                        "Name": "Developer Test Account",
                        "TargetId": "333333333333",
                        "Type": "ACCOUNT"
                },
                {
                        "Arn":"arn:aws:organizations::111111111111:ou/o-
exampleorgid/ou-examplerootid111-exampleouid111",
                        "Name":"Accounting",
                        "TargetId":"ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTargetsForPolicyà la section Référence des AWS CLI commandes.

`move-account`

L'exemple de code suivant montre comment utiliser`move-account`.

AWS CLI

Pour déplacer un compte entre les racines ou OUs

L'exemple suivant montre comment déplacer le compte principal de l'organisation de la racine vers une unité d'organisation :

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-examplerootid111 --destination-parent-id ou-examplerootid111-exampleouid111
```

- Pour plus de détails sur l'API, reportez-vous MoveAccountà la section Référence des AWS CLI commandes.

`remove-account-from-organization`

L'exemple de code suivant montre comment utiliser`remove-account-from-organization`.

AWS CLI

Pour supprimer un compte d'une organisation en tant que compte principal

L'exemple suivant montre comment supprimer un compte d'une organisation :

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- Pour plus de détails sur l'API, reportez-vous RemoveAccountFromOrganizationà la section Référence des AWS CLI commandes.

`update-organizational-unit`

L'exemple de code suivant montre comment utiliser`update-organizational-unit`.

AWS CLI

Pour renommer une UO

Cet exemple vous montre comment renommer une unité d'organisation : dans cet exemple, l'unité d'organisation est renommée « AccountingOU » :

```
aws organizations update-organizational-unit --organizational-unit-id ou-examplerootid111-exampleouid111 --name AccountingOU
```

Le résultat indique le nouveau nom :

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111"
                "Name": "AccountingOU",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-examplerootid111-exampleouid111""
        }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateOrganizationalUnitàla section Référence des AWS CLI commandes.

update-policy

L'exemple de code suivant montre comment utiliserupdate-policy.

AWS CLI

Exemple 1 : pour renommer une politique

L'update-policyexemple suivant renomme une politique et lui donne une nouvelle description.

```
aws organizations update-policy \
    --policy-id p-examplepolicyid111 \
    --name Renamed-Policy \
    --description "This description replaces the original."
```

La sortie indique le nouveau nom et la nouvelle description.

```
{
    "Policy": {
        "Content": "{\n  \"Version\":\"2012-10-17\",\n  \"Statement\":{\n
  \"Effect\":\"Allow\",\n    \"Action\":\"ec2:*\",\n    \"Resource\":\"*\"\n
  }\n}\n",
        "PolicySummary": {
            "Id": "p-examplepolicyid111",
            "AwsManaged": false,
            "Arn":"arn:aws:organizations::111111111111:policy/o-exampleorgid/
service_control_policy/p-examplepolicyid111",
            "Description": "This description replaces the original.",
            "Name": "Renamed-Policy",
            "Type": "SERVICE_CONTROL_POLICY"
        }
    }
}
```

Exemple 2 : pour remplacer le contenu textuel JSON d'une politique

L'exemple suivant vous montre comment remplacer le texte JSON du SCP dans l'exemple précédent par une nouvelle chaîne de texte de politique JSON qui autorise S3 au lieu de EC2 :

```
aws organizations update-policy \
    --policy-id p-examplepolicyid111 \
    --content "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",
\"Action\":\"s3:*\",\"Resource\":\"*\"}}"
```

La sortie montre le nouveau contenu :

```
{
    "Policy": {
        "Content": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
 \"Allow\", \"Action\": \"s3:*\", \"Resource\": \"*\" } }",
        "PolicySummary": {
            "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/
service_control_policy/p-examplepolicyid111",
            "AwsManaged": false;
            "Description": "This description replaces the original.",
            "Id": "p-examplepolicyid111",
            "Name": "Renamed-Policy",
            "Type": "SERVICE_CONTROL_POLICY"
        }
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdatePolicyà la section Référence des AWS CLI commandes.

AWS Outposts exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Outposts.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-outpost-instance-types`

L'exemple de code suivant montre comment utiliser`get-outpost-instance-types`.

AWS CLI

Pour obtenir les types d'instances de votre Outpost

L'`get-outpost-instance-types`exemple suivant obtient les types d'instances pour l'Outpost spécifié.

```
aws outposts get-outpost-instance-types \
    --outpost-id op-0ab23c4567EXAMPLE
```

Sortie :

```
{
```

```
    "InstanceTypes": [
        {
            "InstanceType": "c5d.large"
        },
        {
            "InstanceType": "i3en.24xlarge"
        },
        {
            "InstanceType": "m5d.large"
        },
        {
            "InstanceType": "r5d.large"
        }
    ],
    "OutpostId": "op-0ab23c4567EXAMPLE",
    "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE"
}
```

Pour plus d'informations, consultez la section Lancer une instance sur votre Outpost dans le Guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous GetOutpostInstanceTypesà la section Référence des AWS CLI commandes.

`get-outpost`

L'exemple de code suivant montre comment utiliser`get-outpost`.

AWS CLI

Pour obtenir des informations sur Outpost

L'`get-outpost`exemple suivant affiche les détails de l'avant-poste spécifié.

```
aws outposts get-outpost \
    --outpost-id op-0ab23c4567EXAMPLE
```

Sortie :

```
{
    "Outpost": {
```

```
        "OutpostId": "op-0ab23c4567EXAMPLE",
        "OwnerId": "123456789012",
        "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",
        "SiteId": "os-0ab12c3456EXAMPLE",
        "Name": "EXAMPLE",
        "LifeCycleStatus": "ACTIVE",
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az1",
        "Tags": {}
    }
}
```

Pour plus d'informations, consultez la section Travailler avec les Outposts dans le guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous GetOutpostà la section Référence des AWS CLI commandes.

`list-outposts`

L'exemple de code suivant montre comment utiliser`list-outposts`.

AWS CLI

Pour répertorier les Outposts

L'`list-outposts`exemple suivant répertorie les Outposts de votre AWS compte.

```
aws outposts list-outposts
```

Sortie :

```
{
    "Outposts": [
        {
            "OutpostId": "op-0ab23c4567EXAMPLE",
            "OwnerId": "123456789012",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",
            "SiteId": "os-0ab12c3456EXAMPLE",
            "Name": "EXAMPLE",
            "Description": "example",
```

```
            "LifeCycleStatus": "ACTIVE",
            "AvailabilityZone": "us-west-2a",
            "AvailabilityZoneId": "usw2-az1",
            "Tags": {
                "Name": "EXAMPLE"
            }
        },
        {
            "OutpostId": "op-4fe3dc21baEXAMPLE",
            "OwnerId": "123456789012",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-4fe3dc21baEXAMPLE",
            "SiteId": "os-0ab12c3456EXAMPLE",
            "Name": "EXAMPLE2",
            "LifeCycleStatus": "ACTIVE",
            "AvailabilityZone": "us-west-2a",
            "AvailabilityZoneId": "usw2-az1",
            "Tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec les Outposts dans le guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous ListOutpostsà la section Référence des AWS CLI commandes.

`list-sites`

L'exemple de code suivant montre comment utiliser`list-sites`.

AWS CLI

Pour répertorier des sites

L'`list-sites`exemple suivant répertorie les sites Outpost disponibles dans votre AWS compte.

```
aws outposts list-sites
```

Sortie :

```
{
```

```
    "Sites": [
        {
            "SiteId": "os-0ab12c3456EXAMPLE",
            "AccountId": "123456789012",
            "Name": "EXAMPLE",
            "Description": "example",
            "Tags": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec les Outposts dans le guide de l'utilisateur d'AWS Outposts.

- Pour plus de détails sur l'API, reportez-vous ListSitesà la section Référence des AWS CLI commandes.

AWS Payment Cryptography exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Payment Cryptography.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-alias`

L'exemple de code suivant montre comment utiliser`create-alias`.

AWS CLI

Pour créer un alias pour une clé

L'`create-alias`exemple suivant crée un alias pour une clé.

```
aws payment-cryptography create-alias \
    --alias-name alias/sampleAlias1 \
    --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
    }
}
```

Pour plus d'informations, consultez la section À propos des alias dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir CreateAliasla section Référence des AWS CLI commandes.

`create-key`

L'exemple de code suivant montre comment utiliser`create-key`.

AWS CLI

Pour créer une clé

L'`create-key`exemple suivant génère une clé TDES 2KEY que vous pouvez utiliser pour générer et vérifier des valeurs CVV2 CVV/.

```
aws payment-cryptography create-key \
    --exportable \
    --key-
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC
```

Sortie :

```
{
    "Key": {
```

```
        "CreateTimestamp": "1686800690",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "1686800690"
    }
}
```

Pour plus d'informations, consultez la section Génération de clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir CreateKeyla section Référence des AWS CLI commandes.

`delete-alias`

L'exemple de code suivant montre comment utiliser`delete-alias`.

AWS CLI

Pour supprimer un alias

L'`delete-alias`exemple suivant supprime un alias. Cela n'affecte pas la clé.

```
aws payment-cryptography delete-alias \
    --alias-name alias/sampleAlias1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section À propos des alias dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir DeleteAliasla section Référence des AWS CLI commandes.

`delete-key`

L'exemple de code suivant montre comment utiliser`delete-key`.

AWS CLI

Pour supprimer une clé

L'`delete-key`exemple suivant planifie la suppression d'une clé après 7 jours, qui est la période d'attente par défaut.

```
aws payment-cryptography delete-key \
    --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "Key": {
        "CreateTimestamp": "1686801198",
        "DeletePendingTimestamp": "1687405998",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
```

```
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "1686801190"
    }
}
```

Pour plus d'informations, consultez la section Suppression de clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir DeleteKeyla section Référence des AWS CLI commandes.

`export-key`

L'exemple de code suivant montre comment utiliser`export-key`.

AWS CLI

Pour exporter une clé

L'`export-key`exemple suivant exporte une clé.

```
aws payment-cryptography export-key \
    --export-key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/lco3w6agsk7zgu2l \
    --key-material '{"Tr34KeyBlock": { \
        "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-west-2:123456789012:key/ftobshq7pvioc5fx", \
        "ExportToken": "export-token-cu4lg26ofcziixny", \
        "KeyBlockFormat": "X9_TR34_2012", \
        "WrappingKeyCertificate": file://wrapping-key-certificate.pem }}'
```

Contenu de `wrapping-key-certificate.pem` :

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ0lSQU1ZZS8xMXFUK2svVzlRUDJQOElv
```

Sortie :

```
{
    "WrappedKey": {
        "KeyMaterial":
 "308205A106092A864886F70D010702A08205923082058E020101310D300B0609608648016503040201308203F
        "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
    }
}
```

Pour plus d'informations, consultez la section Exporter des clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir ExportKeyla section Référence des AWS CLI commandes.

`get-alias`

L'exemple de code suivant montre comment utiliser`get-alias`.

AWS CLI

Pour obtenir un alias

L'`get-alias`exemple suivant renvoie l'ARN de la clé associée à l'alias.

```
aws payment-cryptography get-alias \
    --alias-name alias/sampleAlias1
```

Sortie :

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
    }
```

```
}
```

Pour plus d'informations, consultez la section À propos des alias dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir GetAliasla section Référence des AWS CLI commandes.

`get-key`

L'exemple de code suivant montre comment utiliser`get-key`.

AWS CLI

Pour obtenir les métadonnées d'une clé

L'`get-key`exemple suivant renvoie les métadonnées de la clé associée à l'alias. Cette opération ne renvoie aucun matériel cryptographique.

```
aws payment-cryptography get-key \
    --key-identifier alias/sampleAlias1
```

Sortie :

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "DeletePendingTimestamp": "1687405998",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
```

```
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "1686801190"
    }
}
```

Pour plus d'informations, consultez la section Obtenir des clés dans le guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir GetKeyla section Référence des AWS CLI commandes.

`get-parameters-for-export`

L'exemple de code suivant montre comment utiliser`get-parameters-for-export`.

AWS CLI

Pour initialiser le processus d'exportation

L'`get-parameters-for-export`exemple suivant génère une paire de clés, signe la clé, puis renvoie le certificat et la racine du certificat.

```
aws payment-cryptography get-parameters-for-export \
    --signing-key-algorithm RSA_2048 \
    --key-material-type TR34_KEY_BLOCK
```

Sortie :

```
{
    "ExportToken": "export-token-ep5cwyzune7oya53",
    "ParametersValidUntilTimestamp": "1687415640",
    "SigningKeyAlgorithm": "RSA_2048",
    "SigningKeyCertificate":

    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
```

```
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "SigningKeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

Pour plus d'informations, consultez la section Exporter des clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir GetParametersForExportla section Référence des AWS CLI commandes.

`get-parameters-for-import`

L'exemple de code suivant montre comment utiliser`get-parameters-for-import`.

AWS CLI

Pour initialiser le processus d'importation

L'`get-parameters-for-import`exemple suivant génère une paire de clés, signe la clé, puis renvoie le certificat et la racine du certificat.

```
aws payment-cryptography get-parameters-for-import \
    --key-material-type TR34_KEY_BLOCK \
    --wrapping-key-algorithm RSA_2048
```

Sortie :

```
{
    "ImportToken": "import-token-qgmafpaa7nt2kfbb",
    "ParametersValidUntilTimestamp": "1687415640",
    "WrappingKeyAlgorithm": "RSA_2048",
    "WrappingKeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "WrappingKeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
```

```
}
```

Pour plus d'informations, consultez la section Importer des clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir GetParametersForImportla section Référence des AWS CLI commandes.

`get-public-key-certificate`

L'exemple de code suivant montre comment utiliser`get-public-key-certificate`.

AWS CLI

Pour renvoyer la clé publique

L'`get-public-key-certificate`exemple suivant renvoie la partie clé publique d'une paire de clés.

```
aws payment-cryptography get-public-key-certificate \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "KeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "KeyCertificateChain":
```

```
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

Pour plus d'informations, consultez la section key/certificate Associer le public à une paire de clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir GetPublicKeyCertificatela section Référence des AWS CLI commandes.

`import-key`

L'exemple de code suivant montre comment utiliser`import-key`.

AWS CLI

Pour importer une clé TR-34

L'`import-key`exemple suivant importe une clé TR-34.

```
aws payment-cryptography import-key \
        --key-material='{ "Tr34KeyBlock": {" \
            CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564njnjm", \
            "ImportToken": "import-token-5ott6ho5nts7bbcg", \
            "KeyBlockFormat": "X9_TR34_2012", \
            "SigningKeyCertificate": file://signing-key-certificate.pem, \
            "WrappedKeyBlock": file://wrapped-key-block.pem }}'
```

Contenu de `signing-key-certificate.pem` :

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ0lRYWVCK25IbE1WZU1PR1ZiNjU1Q2Jz
```

Contenu de `wrapped-key-block.pem` :

```
3082059806092A864886F70D010702A0820589308205850201013110D300B0609608648016503040201308203160
```

Sortie :

```
{
    "Key": {
        "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
        "Enabled": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
bzmvgyxdg3sktwxd",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "D9B20E",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "EXTERNAL",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2023-06-09T16:56:27.621000-07:00"
    }
}
```

Pour plus d'informations, consultez la section Importer des clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir ImportKeyla section Référence des AWS CLI commandes.

`list-aliases`

L'exemple de code suivant montre comment utiliser`list-aliases`.

AWS CLI

Pour obtenir une liste d'alias

L'`list-aliases`exemple suivant montre tous les alias de votre compte dans cette région.

```
aws payment-cryptography list-aliases
```

Sortie :

```
{
    "Aliases": [
        {
            "AliasName": "alias/sampleAlias1",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h"
        },
        {
            "AliasName": "alias/sampleAlias2",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h"
        }
    ]
}
```

Pour plus d'informations, consultez la section À propos des alias dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir ListAliasesla section Référence des AWS CLI commandes.

`list-keys`

L'exemple de code suivant montre comment utiliser`list-keys`.

AWS CLI

Pour obtenir une liste de clés

L'`list-keys`exemple suivant montre toutes les clés de votre compte dans cette région.

```
aws payment-cryptography list-keys
```

Sortie :

```
{
    "Keys": [
    {
        "CreateTimestamp": "1666506840",
        "Enabled": false,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStopTimestamp": "1666938840"
    }
    ]
}
```

Pour plus d'informations, voir Répertorier les clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir ListKeysla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour obtenir la liste des balises associées à une clé

L'`list-tags-for-resource`exemple suivant permet d'obtenir les balises d'une clé.

```
aws payment-cryptography list-tags-for-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "BIN",
            "Value": "20151120"
        },
        {
            "Key": "Project",
            "Value": "Production"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des tags clés avec les opérations d'API dans le Guide de l'utilisateur AWS de Payment Cryptography.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`restore-key`

L'exemple de code suivant montre comment utiliser`restore-key`.

AWS CLI

Pour restaurer une clé dont la suppression est prévue

L'`restore-key`exemple suivant annule la suppression d'une clé.

```
aws payment-cryptography restore-key \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_3KEY",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
                "Wrap": false,
                "Unwrap": false,
                "Generate": true,
                "Sign": false,
                "Verify": true,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": false,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "1686800690",
        "UsageStopTimestamp": "1687405998"
    }
}
```

Pour plus d'informations, consultez la section Suppression de clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir RestoreKeyla section Référence des AWS CLI commandes.

`start-key-usage`

L'exemple de code suivant montre comment utiliser`start-key-usage`.

AWS CLI

Pour activer une clé

L'`start-key-usage`exemple suivant permet d'utiliser une clé.

```
aws payment-cryptography start-key-usage \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
```

```
        "UsageStartTimestamp": "1686800690"
    }
}
```

Pour plus d'informations, consultez la section Activation et désactivation des clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir StartKeyUsagela section Référence des AWS CLI commandes.

`stop-key-usage`

L'exemple de code suivant montre comment utiliser`stop-key-usage`.

AWS CLI

Pour désactiver une clé

L'`stop-key-usage`exemple suivant désactive une clé.

```
aws payment-cryptography stop-key-usage \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Sortie :

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
```

```
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "1686800690"
    }
}
```

Pour plus d'informations, consultez la section Activation et désactivation des clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir StopKeyUsagela section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter une clé

L'`tag-resource`exemple suivant balise une clé.

```
aws payment-cryptography tag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --tags Key=sampleTag,Value=sampleValue
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des tags clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une étiquette d'une clé

L'`untag-resource`exemple suivant supprime une balise d'une clé.

```
aws payment-cryptography untag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --tag-keys sampleTag
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des tags clés dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-alias`

L'exemple de code suivant montre comment utiliser`update-alias`.

AWS CLI

Pour mettre à jour un alias

L'`update-alias`exemple suivant associe l'alias à une autre clé.

```
aws payment-cryptography update-alias \
    --alias-name alias/sampleAlias1 \
    --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
tqv5yij6wtxx64pi
```

Sortie :

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
tqv5yij6wtxx64pi "
    }
}
```

Pour plus d'informations, consultez la section À propos des alias dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, voir UpdateAliasla section Référence des AWS CLI commandes.

AWS Payment Cryptography Exemples de plans de données utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du plan de AWS Payment Cryptography données AWS Command Line Interface with.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`decrypt-data`

L'exemple de code suivant montre comment utiliser`decrypt-data`.

AWS CLI

Pour déchiffrer le texte chiffré

L'`decrypt-data`exemple suivant déchiffre des données chiffrées à l'aide d'une clé symétrique. Pour cette opération, la clé doit être `KeyModesOfUse` réglée sur `Decrypt` et `KeyUsage` définie sur`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`.

```
aws payment-cryptography-data decrypt-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --cipher-text 33612AB9D6929C3A828EB6030082B2BD \
    --decryption-attributes 'Symmetric={Mode=CBC}'
```

Sortie :

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "71D7AE",
    "PlainText": "31323334313233343132333431323334"
}
```

Pour plus d'informations, consultez la section Déchiffrer les données dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous DecryptDataà la section Référence des AWS CLI commandes.

`encrypt-data`

L'exemple de code suivant montre comment utiliser`encrypt-data`.

AWS CLI

Pour chiffrer des données

L'`encrypt-data`exemple suivant chiffre des données en texte brut à l'aide d'une clé symétrique. Pour cette opération, la clé doit être `KeyModesOfUse` réglée sur `Encrypt` et `KeyUsage` définie sur`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`.

```
aws payment-cryptography-data encrypt-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
```

```
    --plain-text 31323334313233343132333431323334 \
    --encryption-attributes 'Symmetric={Mode=CBC}'
```

Sortie :

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "71D7AE",
    "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

Pour plus d'informations, consultez la section Chiffrer les données dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous EncryptDataà la section Référence des AWS CLI commandes.

`generate-card-validation-data`

L'exemple de code suivant montre comment utiliser`generate-card-validation-data`.

AWS CLI

Pour générer un CVV

L'`generate-card-validation-data`exemple suivant génère un CVV/CVV2.

```
aws payment-cryptography-data generate-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --primary-account-number=171234567890123 \
    --generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

Sortie :

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "CADDA1",
```

```
    "ValidationData": "801"
}
```

Pour plus d'informations, consultez la section Générer des données de carte dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous GenerateCardValidationDataà la section Référence des AWS CLI commandes.

`generate-mac`

L'exemple de code suivant montre comment utiliser`generate-mac`.

AWS CLI

Pour générer un MAC

L'`generate-card-validation-data`exemple suivant génère un code d'authentification de message basé sur le hachage (HMAC) pour l'authentification des données de carte à l'aide de l'algorithme HMAC_ SHA256 et d'une clé de chiffrement HMAC. La clé doit être `KeyUsage` réglée sur `TR31_M7_HMAC_KEY` et `KeyModesOfUse` sur`Generate`.

```
aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --message-
data "3b31303838343930303130373333934313533d323430383232363030303730303033f33" \
    --generation-attributes Algorithm=HMAC_SHA256
```

Sortie :

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h,
    "KeyCheckValue": "2976E7",
    "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
}
```

Pour plus d'informations, consultez la section Générer un MAC dans le guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous GenerateMacà la section Référence des AWS CLI commandes.

`generate-pin-data`

L'exemple de code suivant montre comment utiliser`generate-pin-data`.

AWS CLI

Pour générer un code PIN

L'`generate-card-validation-data`exemple suivant génère un nouveau code PIN aléatoire à l'aide du schéma de code PIN Visa.

```
aws payment-cryptography-data generate-pin-data \
    --generation-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2 \
    --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --primary-account-number 171234567890123 \
    --pin-block-format ISO_FORMAT_0 \
    --generation-attributes VisaPin={PinVerificationKeyIndex=1}
```

Sortie :

```
{
    "GenerationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
    "GenerationKeyCheckValue": "7F2363",
    "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "EncryptionKeyCheckValue": "7CC9E2",
    "EncryptedPinBlock": "AC17DC148BDA645E",
    "PinData": {
        "VerificationValue": "5507"
    }
}
```

Pour plus d'informations, consultez la section Générer des données PIN dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous GeneratePinDataà la section Référence des AWS CLI commandes.

`re-encrypt-data`

L'exemple de code suivant montre comment utiliser`re-encrypt-data`.

AWS CLI

Pour rechiffrer les données avec une autre clé

L'`re-encrypt-data`exemple suivant déchiffre le texte chiffré à l'aide d'une clé symétrique AES et le chiffre à nouveau à l'aide d'une clé unique dérivée par transaction (DUKPT).

```
aws payment-cryptography-data re-encrypt-data \
    --incoming-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/hyvv7ymboitd4vfy \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/jl6ythkcvzesbxen \
    --cipher-
text 4D2B0BDBA192D5AEFEAA5B3EC28E4A65383C313FFA25140101560F75FE1B99F27192A90980AB9334
 \
    --incoming-encryption-
attributes "Dukpt={Mode=ECB,KeySerialNumber=0123456789111111}" \
    --outgoing-encryption-attributes '{"Symmetric": {"Mode": "ECB"}}'
```

Sortie :

```
{
    "CipherText":
 "F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
    "KeyCheckValue": "2E8CD9"
}
```

Pour plus d'informations, consultez la section Chiffrer et déchiffrer des données dans le Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous ReEncryptDataà la section Référence des AWS CLI commandes.

`translate-pin-data`

L'exemple de code suivant montre comment utiliser`translate-pin-data`.

AWS CLI

Pour traduire les données du code PIN

L'`translate-pin-data`exemple suivant traduit un code PIN issu du chiffrement PEK TDES utilisant un bloc PIN ISO 0 en un bloc PIN AES ISO 4 à l'aide de l'algorithme DUKPT.

```
aws payment-cryptography-data translate-pin-data \
    --encrypted-pin-block "AC17DC148BDA645E" \
    --incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}' \
    --incoming-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/4pmyquwjs3yj4vwe \
    --outgoing-translation-attributes
 IsoFormat4="{PrimaryAccountNumber=171234567890123}" \
    --outgoing-dukpt-attributes KeySerialNumber="FFFF9876543210E00008"
```

Sortie :

```
{
    "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt
    "KeyCheckValue": "7CC9E2"
}
```

Pour plus d'informations, consultez la section Translate PIN du guide de l'utilisateur AWS de Payment Cryptography.

- Pour plus de détails sur l'API, reportez-vous TranslatePinDataà la section Référence des AWS CLI commandes.

`verify-auth-request-cryptogram`

L'exemple de code suivant montre comment utiliser`verify-auth-request-cryptogram`.

AWS CLI

Pour vérifier une demande d'authentification

L'`verify-auth-request-cryptogram`exemple suivant vérifie un cryptogramme de demande d'autorisation (ARQC).

```
aws payment-cryptography-data verify-auth-request-cryptogram \
    --auth-request-cryptogram F6E1BD1E6037FB3E \
    --auth-response-attributes '{"ArpcMethod1": {"AuthResponseCode": "1111"}}' \
    --key-identifier arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya \
    --major-key-derivation-mode "EMV_OPTION_A" \
    --session-key-derivation-attributes '{"EmvCommon":
 {"ApplicationTransactionCounter": "1234","PanSequenceNumber":
 "01","PrimaryAccountNumber": "471234567890123"}}' \
    --transaction-data "123456789ABCDEF"
```

Sortie :

```
{
    "AuthResponseValue": "D899B8C6FBF971AA",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya",
    "KeyCheckValue": "985792"
}
```

Pour plus d'informations, consultez la section Cryptogramme ARQC (Verify Auth Request) du Guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous VerifyAuthRequestCryptogramà la section Référence des AWS CLI commandes.

`verify-card-validation-data`

L'exemple de code suivant montre comment utiliser`verify-card-validation-data`.

AWS CLI

Pour valider un CVV

L'`verify-card-validation-data`exemple suivant valide un CVV/ CVV2 pour un PAN.

```
aws payment-cryptography-data verify-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
    --primary-account-number=171234567890123 \
    --verification-attributes CardVerificationValue2={CardExpiryDate=0123} \
    --validation-data 801
```

Sortie :

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "CADDA1"
}
```

Pour plus d'informations, consultez la section Vérifier les données de carte dans le guide de l'utilisateur AWS de Payment Cryptography.

- Pour plus de détails sur l'API, reportez-vous VerifyCardValidationDataà la section Référence des AWS CLI commandes.

`verify-mac`

L'exemple de code suivant montre comment utiliser`verify-mac`.

AWS CLI

Pour vérifier un MAC

L'`verify-mac`exemple suivant vérifie un code d'authentification de message basé sur le hachage (HMAC) pour l'authentification des données de carte à l'aide de l'algorithme HMAC_ SHA256 et d'une clé de chiffrement HMAC.

```
aws payment-cryptography-data verify-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
    --message-
data "3b343038383439303031303733393431353d323430383232363030303730303f33" \
    --verification-attributes='Algorithm=HMAC_SHA256' \
    --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

Sortie :

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6,
    "KeyCheckValue": "2976E7",
}
```

Pour plus d'informations, consultez la section Vérifier le MAC dans le guide de l'utilisateur AWS de Payment Cryptography.

- Pour plus de détails sur l'API, reportez-vous VerifyMacà la section Référence des AWS CLI commandes.

`verify-pin-data`

L'exemple de code suivant montre comment utiliser`verify-pin-data`.

AWS CLI

Pour vérifier un code PIN

L'`verify-pin-data`exemple suivant valide le code PIN d'un PAN.

```
aws payment-cryptography-data verify-pin-data \
    --verification-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2 \
    --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --primary-account-number 171234567890123 \
    --pin-block-format ISO_FORMAT_0 \
    --verification-attributes
 VisaPin="{PinVerificationKeyIndex=1,VerificationValue=5507}" \
    --encrypted-pin-block AC17DC148BDA645E
```

Sortie :

```
{
    "VerificationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
    "VerificationKeyCheckValue": "7F2363",
```

```
    "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "EncryptionKeyCheckValue": "7CC9E2",
}
```

Pour plus d'informations, consultez la section Vérifier les données du code PIN dans le guide de l'utilisateur de la cryptographie des AWS paiements.

- Pour plus de détails sur l'API, reportez-vous VerifyPinDataà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Pinpoint AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Pinpoint.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-app`

L'exemple de code suivant montre comment utiliser`create-app`.

AWS CLI

Exemple 1 : pour créer une application

L’exemple `create-app` suivant crée une application (projet).

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

Exemple 2 : pour créer une application balisée

L'exemple `create-app` suivant crée une application (projet) et associe une balise (clé et valeur) à l'application.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAppà la section Référence des AWS CLI commandes.

`create-sms-template`

L'exemple de code suivant montre comment utiliser`create-sms-template`.

AWS CLI

Crée un modèle de message pour les messages envoyés via le canal SMS

L'`create-sms-template`exemple suivant crée un modèle de message SMS.

```
aws pinpoint create-sms-template \
    --template-name TestTemplate \
    --sms-template-request file://myfile.json \
    --region us-east-1
```

Contenu de `myfile.json` :

```
{
    "Body": "hello\n how are you?\n food is good",
    "TemplateDescription": "Test SMS Template"
}
```

Sortie :

```
{
    "CreateTemplateMessageBody": {
        "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
        "Message": "Created",
        "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"
    }
}
```

Pour plus d'informations, consultez les modèles de messages Amazon Pinpoint dans le guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous CreateSmsTemplateà la section Référence des AWS CLI commandes.

`delete-app`

L'exemple de code suivant montre comment utiliser`delete-app`.

AWS CLI

Pour supprimer une application

L'exemple `delete-app` suivant supprime une application (projet).

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAppà la section Référence des AWS CLI commandes.

`get-apns-channel`

L'exemple de code suivant montre comment utiliser`get-apns-channel`.

AWS CLI

Pour récupérer des informations sur l'état et les paramètres du APNs canal pour une application

L'`get-apns-channel`exemple suivant permet de récupérer des informations sur le statut et les paramètres du APNs canal pour une application.

```
aws pinpoint get-apns-channel \
    --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \
    --region us-east-1
```

Sortie :

```
{
    "APNSChannelResponse": {
        "ApplicationId": "9ab1068eb0a6461c86cce7f27ce0efd7",
        "CreationDate": "2019-05-09T21:54:45.082Z",
        "DefaultAuthenticationMethod": "CERTIFICATE",
```

```
        "Enabled": true,
        "HasCredential": true,
        "HasTokenKey": false,
        "Id": "apns",
        "IsArchived": false,
        "LastModifiedDate": "2019-05-09T22:04:01.067Z",
        "Platform": "APNS",
        "Version": 2
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetApnsChannelà la section Référence des AWS CLI commandes.

`get-app`

L'exemple de code suivant montre comment utiliser`get-app`.

AWS CLI

Pour récupérer des informations sur une application (projet)

L'`get-app`exemple suivant permet de récupérer des informations sur une application (projet).

```
aws pinpoint get-app \
    --application-id 810c7aab86d42fb2b56c8c966example \
    --region us-east-1
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
                "Year": "2019",
                "Stack": "Production"
            }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetAppà la section Référence des AWS CLI commandes.

`get-apps`

L'exemple de code suivant montre comment utiliser`get-apps`.

AWS CLI

Pour récupérer des informations sur toutes vos applications

L'`get-apps`exemple suivant permet de récupérer des informations sur toutes vos applications (projets).

```
aws pinpoint get-apps
```

Sortie :

```
{
    "ApplicationsResponse": {
        "Item": [
            {
                "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
                "Id": "810c7aab86d42fb2b56c8c966example",
                "Name": "ExampleCorp",
                "tags": {
                    "Year": "2019",
                    "Stack": "Production"
                }
            },
            {
                "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/42d8c7eb0990a57ba1d5476a3example",
                "Id": "42d8c7eb0990a57ba1d5476a3example",
                "Name": "AnyCompany",
                "tags": {}
            },
            {
                "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/80f5c382b638ffe5ad12376bbexample",
                "Id": "80f5c382b638ffe5ad12376bbexample",
```

```
                "Name": "ExampleCorp_Test",
                "tags": {
                    "Year": "2019",
                    "Stack": "Test"
                }
            }
        ],
        "NextToken":
 "eyJDcmVhdGlvbkRhdGUiOiIyMDE5LTA3LTE2VDE0OjM4OjUzLjkwM1oiLCJBY2NvdW50SWQiOiI1MTIzOTcxODM4Nz
    }
}
```

La présence de la valeur de `NextToken` réponse indique qu'il y a davantage de sorties disponibles. Appelez à nouveau la commande et indiquez cette valeur comme paramètre `NextToken` d'entrée.

- Pour plus de détails sur l'API, reportez-vous GetAppsà la section Référence des AWS CLI commandes.

`get-campaign`

L'exemple de code suivant montre comment utiliser`get-campaign`.

AWS CLI

Pour récupérer des informations sur le statut, la configuration et les autres paramètres d'une campagne

L'`get-campaign`exemple suivant permet de récupérer des informations sur le statut, la configuration et les autres paramètres d'une campagne.

```
aws pinpoint get-campaign \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \
    --region us-east-1
```

Sortie :

```
{
    "CampaignResponse": {
        "AdditionalTreatments": [],
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
```

```
        "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/
a1e63c6cc0eb43ed826ffcc3cc90b30d",
        "CreationDate": "2019-10-08T18:40:16.581Z",
        "Description": " ",
        "HoldoutPercent": 0,
        "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",
        "IsPaused": false,
        "LastModifiedDate": "2019-10-08T18:40:16.581Z",
        "Limits": {
            "Daily": 0,
            "MaximumDuration": 60,
            "MessagesPerSecond": 50,
            "Total": 0
        },
        "MessageConfiguration": {
            "EmailMessage": {
                "FromAddress": "sender@example.com",
                "HtmlBody": "<!DOCTYPE html>\n  <html lang=\"en\">\n    <head>\n
 <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />\n</head>
\n<body>Hello</body>\n</html>",
                "Title": "PinpointDemo"
            }
        },
        "Name": "MyCampaign",
        "Schedule": {
            "IsLocalTime": false,
            "StartTime": "IMMEDIATE",
            "Timezone": "utc"
        },
        "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
        "SegmentVersion": 1,
        "State": {
            "CampaignStatus": "COMPLETED"
        },
        "tags": {},
        "TemplateConfiguration": {},
        "Version": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetCampaignà la section Référence des AWS CLI commandes.

`get-campaigns`

L'exemple de code suivant montre comment utiliser`get-campaigns`.

AWS CLI

Pour récupérer des informations sur le statut, la configuration et les autres paramètres de toutes les campagnes associées à une application

L'`get-campaigns`exemple suivant permet de récupérer des informations sur le statut, la configuration et les autres paramètres de toutes les campagnes associées à une application.

```
aws pinpoint get-campaigns \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "CampaignsResponse": {
        "Item": [
            {
                "AdditionalTreatments": [],
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/
campaigns/7e1280344c8f4a9aa40a00b006fe44f1",
                "CreationDate": "2019-10-08T18:40:22.905Z",
                "Description": " ",
                "HoldoutPercent": 0,
                "Id": "7e1280344c8f4a9aa40a00b006fe44f1",
                "IsPaused": false,
                "LastModifiedDate": "2019-10-08T18:40:22.905Z",
                "Limits": {},
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": "sender@example.com",
                        "HtmlBody": "<!DOCTYPE html>\n    <html lang=\"en
\">\n    <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html;
 charset=utf-8\" />\n</head>\n<body>Hello</body>\n</html>",
                        "Title": "PInpointDemo Test"
                    }
                },
```

```
                "Name": "MyCampaign1",
                "Schedule": {
                    "IsLocalTime": false,
                    "QuietTime": {},
                    "StartTime": "IMMEDIATE",
                    "Timezone": "UTC"
                },
                "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
                "SegmentVersion": 1,
                "State": {
                    "CampaignStatus": "COMPLETED"
                },
                "tags": {},
                "TemplateConfiguration": {},
                "Version": 1
            },
            {
                "AdditionalTreatments": [],
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/
a1e63c6cc0eb43ed826ffcc3cc90b30d",
                "CreationDate": "2019-10-08T18:40:16.581Z",
                "Description": " ",
                "HoldoutPercent": 0,
                "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",
                "IsPaused": false,
                "LastModifiedDate": "2019-10-08T18:40:16.581Z",
                "Limits": {
                    "Daily": 0,
                    "MaximumDuration": 60,
                    "MessagesPerSecond": 50,
                    "Total": 0
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": "sender@example.com",
                        "HtmlBody": "<!DOCTYPE html>\n    <html lang=\"en
\">\n    <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html;
 charset=utf-8\" />\n</head>\n<body>Demo</body>\n</html>",
                        "Title": "PinpointDemo"
                    }
                },
                "Name": "MyCampaign2",
```

```
                "Schedule": {
                    "IsLocalTime": false,
                    "StartTime": "IMMEDIATE",
                    "Timezone": "utc"
                },
                "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
                "SegmentVersion": 1,
                "State": {
                    "CampaignStatus": "COMPLETED"
                },
                "tags": {},
                "TemplateConfiguration": {},
                "Version": 1
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetCampaignsà la section Référence des AWS CLI commandes.

`get-channels`

L'exemple de code suivant montre comment utiliser`get-channels`.

AWS CLI

Pour récupérer des informations sur l'historique et le statut de chaque canal pour une application

L'`get-channels`exemple suivant permet de récupérer des informations sur l'historique et le statut de chaque canal pour une application.

```
aws pinpoint get-channels \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "ChannelsResponse": {
        "Channels": {
            "GCM": {
```

```
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "CreationDate": "2019-10-08T18:28:23.182Z",
                "Enabled": true,
                "HasCredential": true,
                "Id": "gcm",
                "IsArchived": false,
                "LastModifiedDate": "2019-10-08T18:28:23.182Z",
                "Version": 1
            },
            "SMS": {
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "CreationDate": "2019-10-08T18:39:18.511Z",
                "Enabled": true,
                "Id": "sms",
                "IsArchived": false,
                "LastModifiedDate": "2019-10-08T18:39:18.511Z",
                "Version": 1
            },
            "EMAIL": {
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "CreationDate": "2019-10-08T18:27:23.990Z",
                "Enabled": true,
                "Id": "email",
                "IsArchived": false,
                "LastModifiedDate": "2019-10-08T18:27:23.990Z",
                "Version": 1
            },
            "IN_APP": {
                "Enabled": true,
                "IsArchived": false,
                "Version": 0
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetChannelsà la section Référence des AWS CLI commandes.

`get-email-channel`

L'exemple de code suivant montre comment utiliser`get-email-channel`.

AWS CLI

Pour récupérer des informations sur le statut et les paramètres du canal de messagerie d'une application

L'`get-email-channel`exemple suivant permet de récupérer le statut et les paramètres du canal de courrier électronique d'une application.

```
aws pinpoint get-email-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "EmailChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:27:23.990Z",
        "Enabled": true,
        "FromAddress": "sender@example.com",
        "Id": "email",
        "Identity": "arn:aws:ses:us-east-1:AIDACKCEVSQ6C2EXAMPLE:identity/
sender@example.com",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:27:23.990Z",
        "MessagesPerSecond": 1,
        "Platform": "EMAIL",
        "RoleArn": "arn:aws:iam::AIDACKCEVSQ6C2EXAMPLE:role/pinpoint-events",
        "Version": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetEmailChannelà la section Référence des AWS CLI commandes.

`get-endpoint`

L'exemple de code suivant montre comment utiliser`get-endpoint`.

AWS CLI

Pour récupérer des informations sur les paramètres et les attributs d'un point de terminaison spécifique pour une application

L'exemple `get-endpoint` suivant récupère des informations sur les paramètres et les attributs d'un point de terminaison spécifique pour une application

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

Sortie :

```
{
    "EndpointResponse": {
        "Address": "+11234567890",
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "Attributes": {},
        "ChannelType": "SMS",
        "CohortId": "63",
        "CreationDate": "2019-01-28T23:55:11.534Z",
        "EffectiveDate": "2021-08-06T00:04:51.763Z",
        "EndpointStatus": "ACTIVE",
        "Id": "testendpoint",
        "Location": {
            "Country": "USA"
        },
        "Metrics": {
            "SmsDelivered": 1.0
        },
        "OptOut": "ALL",
        "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
        "User": {
            "UserAttributes": {
                "Age": [
                    "24"
                ]
            },
        "UserId": "testuser"
        }
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous GetEndpointà la section Référence des AWS CLI commandes.

`get-gcm-channel`

L'exemple de code suivant montre comment utiliser`get-gcm-channel`.

AWS CLI

Pour récupérer des informations sur l'état et les paramètres du canal GCM pour une application

L'`get-gcm-channel`exemple suivant permet de récupérer des informations sur l'état et les paramètres du canal GCM pour une application.

```
aws pinpoint get-gcm-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "GCMChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:28:23.182Z",
        "Enabled": true,
        "HasCredential": true,
        "Id": "gcm",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:28:23.182Z",
        "Platform": "GCM",
        "Version": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetGcmChannelà la section Référence des AWS CLI commandes.

`get-sms-channel`

L'exemple de code suivant montre comment utiliser`get-sms-channel`.

AWS CLI

Pour récupérer des informations sur le statut et les paramètres du canal SMS pour une application

L'exemple `get-sms-channel` suivant récupère le statut et les paramètres du canal SMS pour une application.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "SMSChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:39:18.511Z",
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSmsChannelà la section Référence des AWS CLI commandes.

`get-sms-template`

L'exemple de code suivant montre comment utiliser`get-sms-template`.

AWS CLI

Récupère le contenu et les paramètres d'un modèle de message pour les messages envoyés via le canal SMS

L'`get-sms-template`exemple suivant extrait le contenu et les paramètres d'un modèle de message SMS.

```
aws pinpoint get-sms-template \
    --template-name TestTemplate \
    --region us-east-1
```

Sortie :

```
{
    "SMSTemplateResponse": {
        "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
        "Body": "hello\n how are you?\n food is good",
        "CreationDate": "2023-06-20T21:37:30.124Z",
        "LastModifiedDate": "2023-06-20T21:37:30.124Z",
        "tags": {},
        "TemplateDescription": "Test SMS Template",
        "TemplateName": "TestTemplate",
        "TemplateType": "SMS",
        "Version": "1"
    }
}
```

Pour plus d'informations, consultez les modèles de messages Amazon Pinpoint dans le guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous GetSmsTemplateà la section Référence des AWS CLI commandes.

`get-voice-channel`

L'exemple de code suivant montre comment utiliser`get-voice-channel`.

AWS CLI

Pour récupérer des informations sur l'état et les paramètres du canal vocal d'une application

L'`get-voice-channel`exemple suivant permet de récupérer le statut et les paramètres du canal vocal d'une application.

```
aws pinpoint get-voice-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "VoiceChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2022-04-28T00:17:03.836Z",
        "Enabled": true,
        "Id": "voice",
        "IsArchived": false,
        "LastModifiedDate": "2022-04-28T00:17:03.836Z",
        "Platform": "VOICE",
        "Version": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetVoiceChannelà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour récupérer une liste de balises pour une ressource

L'`list-tags-for-resource`exemple suivant récupère toutes les balises (noms de clé et valeurs) associées à la ressource spécifiée.

```
aws pinpoint list-tags-for-resource \
    --resource-arn arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

Sortie :

```
{
    "TagsModel": {
        "tags": {
            "Year": "2019",
            "Stack": "Production"
        }
    }
}
```

Pour plus d'informations, consultez « Tagging Amazon Pinpoint Resources https://docs.aws.amazon.com/pinpoint/ latest/developerguide/tagging < -resources.html>'__ » dans le manuel Amazon Pinpoint Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`phone-number-validate`

L'exemple de code suivant montre comment utiliser`phone-number-validate`.

AWS CLI

Récupère les informations relatives à un numéro de téléphone

Ce qui suit `phone-number-validate` permet de récupérer des informations relatives à un numéro de téléphone.

```
aws pinpoint phone-number-validate \
    --number-validate-request PhoneNumber="+12065550142" \
    --region us-east-1
```

Sortie :

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp Mobile",
        "City": "Seattle",
        "CleansedPhoneNumberE164": "+12065550142",
        "CleansedPhoneNumberNational": "2065550142",
        "Country": "United States",
        "CountryCodeIso2": "US",
```

```
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+12065550142",
        "PhoneType": "MOBILE",
        "PhoneTypeCode": 0,
        "Timezone": "America/Los_Angeles",
        "ZipCode": "98101"
    }
}
```

Pour plus d'informations, consultez Canal Amazon Pinpoint SMS dans le Guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous PhoneNumberValidateà la section Référence des AWS CLI commandes.

`send-messages`

L'exemple de code suivant montre comment utiliser`send-messages`.

AWS CLI

Pour envoyer un message SMS à l'aide du point de terminaison d'une application

L'exemple `send-messages` suivant envoie un message direct pour une application dotée d'un point de terminaison.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

Contenu de `myfile.json` :

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
        "testendpoint": {}
```

```
    }
}
```

Sortie :

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

Pour plus d'informations, consultez Canal Amazon Pinpoint SMS dans le Guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section Référence des AWS CLI commandes.

`send-users-messages`

L'exemple de code suivant montre comment utiliser`send-users-messages`.

AWS CLI

Pour envoyer un message SMS à un utilisateur d'une application

L'`send-users-messages`exemple suivant envoie un message direct à un utilisateur d'une application.

```
aws pinpoint send-users-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
```

```
    --send-users-message-request file://myfile.json \
    --region us-west-2
```

Contenu de `myfile.json` :

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Users": {
        "testuser": {}
    }
}
```

Sortie :

```
{
    "SendUsersMessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",
        "Result": {
            "testuser": {
                "testuserendpoint": {
                    "DeliveryStatus": "SUCCESSFUL",
                    "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcuh8b7p45ifsmo0",
                    "StatusCode": 200,
                    "StatusMessage": "MessageId:
 7qu4hk5bqhda3i7i2n4pjf98qcuh8b7p45ifsmo0",
                    "Address": "+12345678900"
                }
            }
        }
    }
}
```

Pour plus d'informations, consultez Canal Amazon Pinpoint SMS dans le Guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous SendUsersMessagesà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une ressource

L'exemple suivant ajoute deux balises (noms de clés et valeurs) à une ressource.

```
aws pinpoint list-tags-for-resource \
    --resource-arn arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tags-model tags={Stack=Production,Year=2019}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez « Tagging Amazon Pinpoint Resources https:// docs.aws.amazon.com/pinpoint/ latest/developerguide/tagging < -resources.html>'__ » dans le manuel Amazon Pinpoint Developer Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Exemple 1 : pour supprimer un tag d'une ressource

L'`untag-resource`exemple suivant supprime la balise spécifiée (nom de clé et valeur) d'une ressource.

```
aws pinpoint untag-resource \
    --resource-arn arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tag-keys Year
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour supprimer plusieurs balises d'une ressource

L'`untag-resource`exemple suivant supprime les balises spécifiées (noms de clés et valeurs) d'une ressource.

```
aws pinpoint untag-resource \
    --resource-arn arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tag-keys Year Stack
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez « Tagging Amazon Pinpoint Resources https:// docs.aws.amazon.com/pinpoint/ latest/developerguide/tagging < -resources.html>'__ » dans le manuel Amazon Pinpoint Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-sms-channel`

L'exemple de code suivant montre comment utiliser`update-sms-channel`.

AWS CLI

Pour activer le canal SMS ou pour mettre à jour le statut et les paramètres du canal SMS pour une application.

L'`update-sms-channel`exemple suivant active le canal SMS pour un canal SMS pour une application.

```
aws pinpoint update-sms-channel \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --sms-channel-request Enabled=true \
    --region us-west-2
```

Sortie :

```
{
    "SMSChannelResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "CreationDate": "2019-01-28T23:25:25.224Z",
```

```
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2023-05-18T23:22:50.977Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 3
    }
}
```

Pour plus d'informations, consultez Canal Amazon Pinpoint SMS dans le Guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous UpdateSmsChannelà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Polly AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Polly.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-lexicon`

L'exemple de code suivant montre comment utiliser`delete-lexicon`.

AWS CLI

Pour supprimer un lexique

L'`delete-lexicon`exemple suivant supprime le lexique spécifié.

```
aws polly delete-lexicon \
    --name w3c
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de l' DeleteLexicon opération dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteLexiconà la section Référence des AWS CLI commandes.

`get-lexicon`

L'exemple de code suivant montre comment utiliser`get-lexicon`.

AWS CLI

Pour récupérer le contenu d'un lexique

L'`get-lexicon`exemple suivant permet de récupérer le contenu du lexique de prononciation spécifié.

```
aws polly get-lexicon \
    --name w3c
```

Sortie :

```
{
    "Lexicon": {
        "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=
\"1.0\" \n      xmlns=    \"http://www.w3.org/2005/01/pronunciation-lexicon
\"\n      xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" \n
 xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon \n
 http://www.w3.org/TR/2007/CR-pronunciation-    lexicon-20071212/pls.xsd\"\n
   alphabet=\"ipa\" \n      xml:lang=\"en-US\">\n  <lexeme>\n    <grapheme>W3C</
grapheme>\n        <alias>World Wide Web Consortium</alias>\n  </lexeme>\n</lexicon>
\n",
        "Name": "w3c"
    },
    "LexiconAttributes": {
```

```
        "Alphabet": "ipa",
        "LanguageCode": "en-US",
        "LastModified": 1603908910.99,
        "LexiconArn": "arn:aws:polly:us-west-2:880185128111:lexicon/w3c",
        "LexemesCount": 1,
        "Size": 492
    }
}
```

Pour plus d'informations, consultez la section Utilisation de l' GetLexicon opération dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetLexiconà la section Référence des AWS CLI commandes.

`get-speech-synthesis-task`

L'exemple de code suivant montre comment utiliser`get-speech-synthesis-task`.

AWS CLI

Pour obtenir des informations sur une tâche de synthèse vocale

L'`get-speech-synthesis-task`exemple suivant permet de récupérer des informations sur la tâche de synthèse vocale spécifiée.

```
aws polly get-speech-synthesis-task \
    --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Sortie :

```
{
    "SynthesisTask": {
        "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
        "TaskStatus": "completed",
        "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
```

```
    }
}
```

Pour plus d'informations, consultez la section Création de longs fichiers audio dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetSpeechSynthesisTaskà la section Référence des AWS CLI commandes.

`list-lexicons`

L'exemple de code suivant montre comment utiliser`list-lexicons`.

AWS CLI

Pour répertorier vos lexiques

L'`list-lexicons`exemple suivant répertorie vos lexiques de prononciation.

```
aws polly list-lexicons
```

Sortie :

```
{
    "Lexicons": [
        {
            "Name": "w3c",
            "Attributes": {
                "Alphabet": "ipa",
                "LanguageCode": "en-US",
                "LastModified": 1603908910.99,
                "LexiconArn": "arn:aws:polly:us-east-2:123456789012:lexicon/w3c",
                "LexemesCount": 1,
                "Size": 492
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de l' ListLexicons opération dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListLexiconsà la section Référence des AWS CLI commandes.

`list-speech-synthesis-tasks`

L'exemple de code suivant montre comment utiliser`list-speech-synthesis-tasks`.

AWS CLI

Pour répertorier vos tâches de synthèse vocale

L'`list-speech-synthesis-tasks`exemple suivant répertorie vos tâches de synthèse vocale.

```
aws polly list-speech-synthesis-tasks
```

Sortie :

```
{
    "SynthesisTasks": [
        {
            "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
            "TaskStatus": "completed",
            "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
            "CreationTime": 1603911042.689,
            "RequestCharacters": 1311,
            "OutputFormat": "mp3",
            "TextType": "text",
            "VoiceId": "Joanna"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création de longs fichiers audio dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListSpeechSynthesisTasksà la section Référence des AWS CLI commandes.

`put-lexicon`

L'exemple de code suivant montre comment utiliser`put-lexicon`.

AWS CLI

Pour enregistrer un lexique

L'`put-lexicon`exemple suivant enregistre le lexique de prononciation spécifié. Le `example.pls` fichier spécifie un lexique conforme au W3C PLS.

```
aws polly put-lexicon \
    --name w3c \
    --content file://example.pls
```

Contenu de `example.pls`

```
{
    <?xml version="1.0" encoding="UTF-8"?>
    <lexicon version="1.0"
        xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
            http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
        alphabet="ipa"
        xml:lang="en-US">
        <lexeme>
            <grapheme>W3C</grapheme>
            <alias>World Wide Web Consortium</alias>
        </lexeme>
    </lexicon>
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de l' PutLexicon opération dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutLexiconà la section Référence des AWS CLI commandes.

`start-speech-synthesis-task`

L'exemple de code suivant montre comment utiliser`start-speech-synthesis-task`.

AWS CLI

Pour synthétiser du texte

L'`start-speech-synthesis-task`exemple suivant synthétise le texte `text_file.txt` et stocke le MP3 fichier obtenu dans le compartiment spécifié.

```
aws polly start-speech-synthesis-task \
    --output-format mp3 \
    --output-s3-bucket-name amzn-s3-demo-bucket \
    --text  file://text_file.txt \
    --voice-id Joanna
```

Sortie :

```
{
    "SynthesisTask": {
        "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
        "TaskStatus": "scheduled",
        "OutputUri": "https://s3.us-east-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

Pour plus d'informations, consultez la section Création de longs fichiers audio dans le manuel Amazon Polly Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartSpeechSynthesisTaskà la section Référence des AWS CLI commandes.

AWS Price List exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Price List.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-services`

L'exemple de code suivant montre comment utiliser`describe-services`.

AWS CLI

Pour récupérer les métadonnées du service

Cet exemple extrait les métadonnées du code de EC2 service Amazon.

Commande :

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --
max-items 1
```

Sortie :

```
{
  "Services": [
      {
          "ServiceCode": "AmazonEC2",
          "AttributeNames": [
              "volumeType",
              "maxIopsvolume",
              "instance",
              "instanceCapacity10xlarge",
              "locationType",
              "instanceFamily",
              "operatingSystem",
              "clockSpeed",
```

```
                "LeaseContractLength",
                "ecu",
                "networkPerformance",
                "instanceCapacity8xlarge",
                "group",
                "maxThroughputvolume",
                "gpuMemory",
                "ebsOptimized",
                "elasticGpuType",
                "maxVolumeSize",
                "gpu",
                "processorFeatures",
                "intelAvxAvailable",
                "instanceCapacity4xlarge",
                "servicecode",
                "groupDescription",
                "processorArchitecture",
                "physicalCores",
                "productFamily",
                "enhancedNetworkingSupported",
                "intelTurboAvailable",
                "memory",
                "dedicatedEbsThroughput",
                "vcpu",
                "OfferingClass",
                "instanceCapacityLarge",
                "capacitystatus",
                "termType",
                "storage",
                "intelAvx2Available",
                "storageMedia",
                "physicalProcessor",
                "provisioned",
                "servicename",
                "PurchaseOption",
                "instanceCapacity18xlarge",
                "instanceType",
                "tenancy",
                "usagetype",
                "normalizationSizeFactor",
                "instanceCapacity2xlarge",
                "instanceCapacity16xlarge",
                "maxIopsBurstPerformance",
                "instanceCapacity12xlarge",
```

```
                "instanceCapacity32xlarge",
                "instanceCapacityXlarge",
                "licenseModel",
                "currentGeneration",
                "preInstalledSw",
                "location",
                "instanceCapacity24xlarge",
                "instanceCapacity9xlarge",
                "instanceCapacityMedium",
                "operation"
            ]
        }
    ],
    "FormatVersion": "aws_v1"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeServicesà la section Référence des AWS CLI commandes.

`get-attribute-values`

L'exemple de code suivant montre comment utiliser`get-attribute-values`.

AWS CLI

Pour récupérer une liste de valeurs d'attributs

L'`get-attribute-values`exemple suivant récupère une liste de valeurs disponibles pour l'attribut donné.

```
aws pricing get-attribute-values \
    --service-code AmazonEC2 \
    --attribute-name volumeType \
    --max-items 2
```

Sortie :

```
{
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",
    "AttributeValues": [
        {
            "Value": "Cold HDD"
```

```
        },
        {
            "Value": "General Purpose"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetAttributeValuesà la section Référence des AWS CLI commandes.

`get-products`

L'exemple de code suivant montre comment utiliser`get-products`.

AWS CLI

Pour récupérer une liste de produits

Cet exemple permet de récupérer une liste de produits correspondant aux critères définis.

Commande :

```
aws pricing get-products --filters file://filters.json --format-version aws_v1 --
max-results 1 --service-code AmazonEC2
```

filtres.json :

```
        [
  {
    "Type": "TERM_MATCH",
    "Field": "ServiceCode",
    "Value": "AmazonEC2"
  },
  {
    "Type": "TERM_MATCH",
    "Field": "volumeType",
    "Value": "Provisioned IOPS"
  }
]
```

Sortie :

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVSagyIFn770XQOzdUIcO9BY6ucBG9itXAZGZF/
zioUzOsUKh6PCcPWaOyPZRiMePb986TeoKYB9l55fw/
CyoMq5ymnGmT1Vj39TljbbAlhcqnVfTmPIilx8Uy5bdDaBYy/e/2Ofw9Edzsykbs8LTBuNbiDQ
+BBds5yeI9AQkUepruKk3aEahFPxJ55kx/zk",
  "PriceList": [
      "{\"product\":{\"productFamily\":\"Storage\",\"attributes\":{\"storageMedia\":
\"SSD-backed\",\"maxThroughputvolume\":\"320 MB/sec\",\"volumeType\":\"Provisioned
 IOPS\",\"maxIopsvolume\":\"20000\",\"servicecode\":\"AmazonEC2\",\"usagetype
\":\"APS1-EBS:VolumeUsage.piops\",\"locationType\":\"AWS Region\",\"location\":
\"Asia Pacific (Singapore)\",\"servicename\":\"Amazon Elastic Compute Cloud\",
\"maxVolumeSize\":\"16 TiB\",\"operation\":\"\"},\"sku\":\"3MKHN58N7RDDVGKJ\"},
\"serviceCode\":\"AmazonEC2\",\"terms\":{\"OnDemand\":{\"3MKHN58N7RDDVGKJ.JRTCKXETXF
\":{\"priceDimensions\":{\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\":{\"unit\":\"GB-
Mo\",\"endRange\":\"Inf\",\"description\":\"$0.138 per GB-month of Provisioned
 IOPS SSD (io1)  provisioned storage - Asia Pacific (Singapore)\",\"appliesTo
\":[],\"rateCode\":\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\",\"beginRange\":
\"0\",\"pricePerUnit\":{\"USD\":\"0.1380000000\"}}},\"sku\":\"3MKHN58N7RDDVGKJ
\",\"effectiveDate\":\"2018-08-01T00:00:00Z\",\"offerTermCode\":\"JRTCKXETXF
\",\"termAttributes\":{}}}},\"version\":\"20180808005701\",\"publicationDate\":
\"2018-08-08T00:57:01Z\"}"
  ]
}
```


- Pour plus de détails sur l'API, reportez-vous GetProductsà la section Référence des AWS CLI commandes.

AWS Private CA exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Private CA.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-certificate-authority-audit-report`

L'exemple de code suivant montre comment utiliser`create-certificate-authority-audit-report`.

AWS CLI

Pour créer un rapport d'audit de l'autorité de certification

La `create-certificate-authority-audit-report` commande suivante crée un rapport d'audit pour l'autorité de certification privée identifiée par l'ARN.

```
aws acm-pca create-certificate-authority-audit-report --certificate-authority-arn arn:aws:acm-pca:us-east-1:accountid:certificate-authority/12345678-1234-1234-1234-123456789012 --s3-bucket-name your-bucket-name --audit-report-response-format JSON
```

- Pour plus de détails sur l'API, reportez-vous CreateCertificateAuthorityAuditReportà la section Référence des AWS CLI commandes.

`create-certificate-authority`

L'exemple de code suivant montre comment utiliser`create-certificate-authority`.

AWS CLI

Pour créer une autorité de certification privée

La `create-certificate-authority` commande suivante crée une autorité de certification privée dans votre AWS compte.

```
aws acm-pca create-certificate-authority --certificate-authority-configuration file://C:\ca_config.txt --revocation-configuration file://C:\revoke_config.txt --certificate-authority-type "SUBORDINATE" --idempotency-token 98256344
```

- Pour plus de détails sur l'API, reportez-vous CreateCertificateAuthorityà la section Référence des AWS CLI commandes.

`delete-certificate-authority`

L'exemple de code suivant montre comment utiliser`delete-certificate-authority`.

AWS CLI

Pour supprimer une autorité de certification privée

La `delete-certificate-authority` commande suivante supprime l'autorité de certification identifiée par l'ARN.

```
aws acm-pca delete-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- Pour plus de détails sur l'API, reportez-vous DeleteCertificateAuthorityà la section Référence des AWS CLI commandes.

`describe-certificate-authority-audit-report`

L'exemple de code suivant montre comment utiliser`describe-certificate-authority-audit-report`.

AWS CLI

Pour décrire un rapport d'audit destiné à une autorité de certification

La `describe-certificate-authority-audit-report` commande suivante répertorie les informations relatives au rapport d'audit spécifié pour l'autorité de certification identifiée par l'ARN.

```
aws acm-pca describe-certificate-authority-audit-report --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/99999999-8888-7777-6666-555555555555 --audit-report-
id 11111111-2222-3333-4444-555555555555
```

- Pour plus de détails sur l'API, reportez-vous DescribeCertificateAuthorityAuditReportà la section Référence des AWS CLI commandes.

`describe-certificate-authority`

L'exemple de code suivant montre comment utiliser`describe-certificate-authority`.

AWS CLI

Pour décrire une autorité de certification privée

La `describe-certificate-authority` commande suivante répertorie les informations relatives à l'autorité de certification privée identifiée par l'ARN.

```
aws acm-pca describe-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- Pour plus de détails sur l'API, reportez-vous DescribeCertificateAuthorityà la section Référence des AWS CLI commandes.

`get-certificate-authority-certificate`

L'exemple de code suivant montre comment utiliser`get-certificate-authority-certificate`.

AWS CLI

Pour récupérer un certificat d'autorité de certification (CA)

La `get-certificate-authority-certificate` commande suivante permet de récupérer le certificat et la chaîne de certificats pour l'autorité de certification privée spécifiée par l'ARN.

```
aws acm-pca get-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --output text
```

- Pour plus de détails sur l'API, reportez-vous GetCertificateAuthorityCertificateà la section Référence des AWS CLI commandes.

`get-certificate-authority-csr`

L'exemple de code suivant montre comment utiliser`get-certificate-authority-csr`.

AWS CLI

Pour récupérer la demande de signature de certificat pour une autorité de certification

La `get-certificate-authority-csr` commande suivante récupère le CSR de l'autorité de certification privée spécifiée par l'ARN.

```
aws acm-pca get-certificate-authority-csr --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --output text
```

- Pour plus de détails sur l'API, reportez-vous GetCertificateAuthorityCsrà la section Référence des AWS CLI commandes.

`get-certificate`

L'exemple de code suivant montre comment utiliser`get-certificate`.

AWS CLI

Pour récupérer un certificat émis

L'`get-certificate`exemple suivant extrait un certificat auprès de l'autorité de certification privée spécifiée.

```
aws acm-pca get-certificate \
    --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 \
    --certificate-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012/
certificate/6707447683a9b7f4055627ffd55cebcc \
    --output text
```

Sortie :

```
-----BEGIN CERTIFICATE-----
MIIEDzCCAvegAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvrZMwDQYJKoZIhvcNAQELBQAw
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBgNVBAcMB1Nl
    ....certificate body truncated for brevity....
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqskTu/KSS7YDjK0
KQNy73D6LtmdOEbAyq10XiDxqY41lvKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE---- -----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIRAOskdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBgNVBAcMB1Nl
    ...certificate body truncated for brevity....
```

```
kdRGB6P2hpxstDOUIwAoCbhoaWwfA4ybJznf+jOQhAziNlRdKQRR8nODWpKt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----
```

La première partie de la sortie est le certificat lui-même. La deuxième partie est la chaîne de certificats qui est liée au certificat de l'autorité de certification racine. Notez que lorsque vous utilisez cette `--output text` option, un TAB caractère est inséré entre les deux éléments du certificat (c'est la cause du texte en retrait). Si vous avez l'intention de prendre cette sortie et d'analyser les certificats avec d'autres outils, vous devrez peut-être supprimer le TAB caractère afin qu'il soit traité correctement.

- Pour plus de détails sur l'API, reportez-vous GetCertificateà la section Référence des AWS CLI commandes.

`import-certificate-authority-certificate`

L'exemple de code suivant montre comment utiliser`import-certificate-authority-certificate`.

AWS CLI

Pour importer votre certificat d'autorité de certification dans ACM PCA

La `import-certificate-authority-certificate` commande suivante importe le certificat de CA privé signé pour l'autorité de certification spécifiée par l'ARN dans ACM PCA.

```
aws acm-pca import-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --certificate file://C:\ca_cert.pem
 --certificate-chain file://C:\ca_cert_chain.pem
```

- Pour plus de détails sur l'API, reportez-vous ImportCertificateAuthorityCertificateà la section Référence des AWS CLI commandes.

`issue-certificate`

L'exemple de code suivant montre comment utiliser`issue-certificate`.

AWS CLI

Pour émettre un certificat privé

La `issue-certificate` commande suivante utilise l'autorité de certification privée spécifiée par l'ARN pour émettre un certificat privé.

```
aws acm-pca issue-certificate --certificate-authority-arn arn:aws:acm-pca:us-
west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012
 --csr file://C:\cert_1.csr --signing-algorithm "SHA256WITHRSA" --validity
 Value=365,Type="DAYS" --idempotency-token 1234
```

- Pour plus de détails sur l'API, reportez-vous IssueCertificateà la section Référence des AWS CLI commandes.

`list-certificate-authorities`

L'exemple de code suivant montre comment utiliser`list-certificate-authorities`.

AWS CLI

Pour répertorier vos autorités de certification privées

La `list-certificate-authorities` commande suivante répertorie les informations relatives à tous les éléments privés CAs de votre compte.

```
aws acm-pca list-certificate-authorities --max-results 10
```

- Pour plus de détails sur l'API, reportez-vous ListCertificateAuthoritiesà la section Référence des AWS CLI commandes.

`list-tags`

L'exemple de code suivant montre comment utiliser`list-tags`.

AWS CLI

Pour répertorier les balises de votre autorité de certification

La `list-tags` commande suivante répertorie les balises associées à l'autorité de certification privée spécifiée par l'ARN.

```
aws acm-pca list-tags --certificate-authority-arn arn:aws:acm-pca:us-
west-2:123456789012:certificate-authority/123455678-1234-1234-1234-123456789012 --
max-results 10
```

- Pour plus de détails sur l'API, reportez-vous ListTagsà la section Référence des AWS CLI commandes.

`revoke-certificate`

L'exemple de code suivant montre comment utiliser`revoke-certificate`.

AWS CLI

Pour révoquer un certificat privé

La `revoke-certificate` commande suivante révoque un certificat privé de l'autorité de certification identifiée par l'ARN.

```
aws acm-pca revoke-certificate --certificate-authority-arn arn:aws:acm-pca:us-
west-2:1234567890:certificate-authority/12345678-1234-1234-1234-123456789012 --
certificate-serial 67:07:44:76:83:a9:b7:f4:05:56:27:ff:d5:5c:eb:cc --revocation-
reason "KEY_COMPROMISE"
```

- Pour plus de détails sur l'API, reportez-vous RevokeCertificateà la section Référence des AWS CLI commandes.

`tag-certificate-authority`

L'exemple de code suivant montre comment utiliser`tag-certificate-authority`.

AWS CLI

Pour associer des balises à une autorité de certification privée

La `tag-certificate-authority` commande suivante attache une ou plusieurs balises à votre autorité de certification privée.

```
aws acm-pca tag-certificate-authority --certificate-authority-
arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice
```

- Pour plus de détails sur l'API, reportez-vous TagCertificateAuthorityà la section Référence des AWS CLI commandes.

`untag-certificate-authority`

L'exemple de code suivant montre comment utiliser`untag-certificate-authority`.

AWS CLI

Pour supprimer une ou plusieurs balises de votre autorité de certification privée

La `untag-certificate-authority` commande suivante supprime les balises de l'autorité de certification privée identifiée par l'ARN.

```
aws acm-pca untag-certificate-authority --certificate-authority-
arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --tags Key=Purpose,Value=Website
```

- Pour plus de détails sur l'API, reportez-vous UntagCertificateAuthorityà la section Référence des AWS CLI commandes.

`update-certificate-authority`

L'exemple de code suivant montre comment utiliser`update-certificate-authority`.

AWS CLI

Pour mettre à jour la configuration de votre autorité de certification privée

La `update-certificate-authority` commande suivante met à jour le statut et la configuration de l'autorité de certification privée identifiée par l'ARN.

```
aws acm-pca update-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-1232456789012 --revocation-configuration file://C:
\revoke_config.txt --status "DISABLED"
```

- Pour plus de détails sur l'API, reportez-vous UpdateCertificateAuthorityà la section Référence des AWS CLI commandes.

AWS Proton exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Proton.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-service-instance-deployment`

L'exemple de code suivant montre comment utiliser`cancel-service-instance-deployment`.

AWS CLI

Pour annuler le déploiement d'une instance de service

L'`cancel-service-instance-deployment`exemple suivant annule le déploiement d'une instance de service.

```
aws proton cancel-service-instance-deployment \
    --service-instance-name "instance-one" \
    --service-name "simple-svc"
```

Sortie :

```
{
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "CANCELLING",
        "environmentName": "simple-env",
        "lastDeploymentAttemptedAt": "2021-04-02T21:45:15.406000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:38:00.823000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_optional_input: abc\n  my_sample_pipeline_required_input:
```

```
 '123'\ninstances:\n- name: my-instance\n  environment: MySimpleEnv
\n  spec:\n    my_sample_service_instance_optional_input: def\n
 my_sample_service_instance_required_input: '456'\n- name: my-other-instance\n
 environment: MySimpleEnv\n  spec:\n    my_sample_service_instance_required_input:
 '789'\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "1",
        "templateName": "svc-simple"
    }
}
```

Pour plus d'informations, voir Mettre à jour une instance de service dans le Guide de l'administrateur de AWS Proton ou Mettre à jour une instance de service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous CancelServiceInstanceDeploymentà la section Référence des AWS CLI commandes.

`cancel-service-pipeline-deployment`

L'exemple de code suivant montre comment utiliser`cancel-service-pipeline-deployment`.

AWS CLI

Pour annuler le déploiement d'un pipeline de services

L'`cancel-service-pipeline-deployment`exemple suivant annule le déploiement d'un pipeline de services.

```
aws proton cancel-service-pipeline-deployment \
    --service-name "simple-svc"
```

Sortie :

```
{
    "pipeline": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "CANCELLING",
        "lastDeploymentAttemptedAt": "2021-04-02T22:02:45.095000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:39:28.991000+00:00",
        "templateMajorVersion": "1",
```

```
        "templateMinorVersion": "1",
        "templateName": "svc-simple"
    }
}
```

Pour plus d'informations, voir Mettre à jour un pipeline de services dans le guide de l'administrateur de AWS Proton ou Mettre à jour un pipeline de services dans le guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous CancelServicePipelineDeploymentà la section Référence des AWS CLI commandes.

`create-service`

L'exemple de code suivant montre comment utiliser`create-service`.

AWS CLI

Pour créer un service

L'`create-service`exemple suivant crée un service avec un pipeline de services.

```
aws proton create-service \
    --name "MySimpleService" \
    --template-name "fargate-service" \
    --template-major-version "1" \
    --branch-name "mainline" \
    --repository-connection-arn "arn:aws:codestar-connections:region-id:account-
id:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
    --repository-id "myorg/myapp" \
    --spec file://spec.yaml
```

Contenu de `spec.yaml` :

```
proton: ServiceSpec

pipeline:
    my_sample_pipeline_required_input: "hello"
    my_sample_pipeline_optional_input: "bye"

instances:
    - name: "acme-network-dev"
```

```
        environment: "ENV_NAME"
        spec:
            my_sample_service_instance_required_input: "hi"
            my_sample_service_instance_optional_input: "ho"
```

Sortie :

```
{
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
        "createdAt": "2020-11-18T19:50:27.460000+00:00",
        "lastModifiedAt": "2020-11-18T19:50:27.460000+00:00",
        "name": "MySimpleService",
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "repositoryId": "myorg/myapp",
        "status": "CREATE_IN_PROGRESS",
        "templateName": "fargate-service"
    }
}
```

Pour plus d'informations, voir Création d'un service dans le Guide de l'administrateur de AWS Proton et Création d'un service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous CreateServiceà la section Référence des AWS CLI commandes.

`delete-service`

L'exemple de code suivant montre comment utiliser`delete-service`.

AWS CLI

Pour supprimer un service

L'`delete-service`exemple suivant supprime un service.

```
aws proton delete-service \
    --name "simple-svc"
```

Sortie :

```
{
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",
        "branchName": "mainline",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "description": "Edit by updating description",
        "lastModifiedAt": "2020-11-29T00:30:39.248000+00:00",
        "name": "simple-svc",
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "repositoryId": "myorg/myapp",
        "status": "DELETE_IN_PROGRESS",
        "templateName": "fargate-service"
    }
}
```

Pour plus d'informations, voir Supprimer un service dans le Guide de l'administrateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceà la section Référence des AWS CLI commandes.

`get-service-instance`

L'exemple de code suivant montre comment utiliser`get-service-instance`.

AWS CLI

Pour obtenir les détails de l'instance de service

L'`get-service-instance`exemple suivant permet d'obtenir des données détaillées pour une instance de service.

```
aws proton get-service-instance \
    --name "instance-one" \
    --service-name "simple-svc"
```

Sortie :

```
{
    "serviceInstance": {
```

```
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "deploymentStatus": "SUCCEEDED",
        "environmentName": "simple-env",
        "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
        "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_optional_input: hello world\n
 my_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\n
 environment: my-simple-env\n  spec:\n    my_sample_service_instance_optional_input:
 Ola\n    my_sample_service_instance_required_input: Ciao\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

Pour plus d'informations, voir Afficher les données de service dans le Guide de l'administrateur de AWS Proton ou Afficher les données de service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous GetServiceInstanceà la section Référence des AWS CLI commandes.

`get-service`

L'exemple de code suivant montre comment utiliser`get-service`.

AWS CLI

Pour obtenir les détails du service

L'`get-service`exemple suivant permet d'obtenir les données détaillées d'un service.

```
aws proton get-service \
    --name "simple-svc"
```

Sortie :

```
{
    "service": {
```

```
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",
        "branchName": "mainline",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "lastModifiedAt": "2020-11-28T22:44:51.207000+00:00",
        "name": "simple-svc",
        "pipeline": {
            "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
pipeline/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "createdAt": "2020-11-28T22:40:50.512000+00:00",
            "deploymentStatus": "SUCCEEDED",
            "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
            "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
            "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_required_input: hello\n  my_sample_pipeline_optional_input:
 bye\ninstances:\n- name: instance-svc-simple\n  environment: my-simple-
env\n  spec:\n    my_sample_service_instance_required_input: hi\n
 my_sample_service_instance_optional_input: ho\n",
            "templateMajorVersion": "1",
            "templateMinorVersion": "1",
            "templateName": "svc-simple"
        },
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "repositoryId": "myorg/myapp",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_required_input: hello\n  my_sample_pipeline_optional_input:
 bye\ninstances:\n- name: instance-svc-simple\n  environment: my-simple-
env\n  spec:\n    my_sample_service_instance_required_input: hi\n
 my_sample_service_instance_optional_input: ho\n",
        "status": "ACTIVE",
        "templateName": "svc-simple"
    }
}
```

Pour plus d'informations, voir Afficher les données de service dans le Guide de l'administrateur de AWS Proton ou Afficher les données de service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous GetServiceà la section Référence des AWS CLI commandes.

`list-service-instances`

L'exemple de code suivant montre comment utiliser`list-service-instances`.

AWS CLI

Exemple 1 : pour répertorier toutes les instances de service

L'`list-service-instances`exemple suivant répertorie les instances de service.

```
aws proton list-service-instances
```

Sortie :

```
{
    "serviceInstances": [
        {
            "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
service-instance/instance-one",
            "createdAt": "2020-11-28T22:40:50.512000+00:00",
            "deploymentStatus": "SUCCEEDED",
            "environmentArn": "arn:aws:proton:region-id:123456789012:environment/
simple-env",
            "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
            "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
            "name": "instance-one",
            "serviceName": "simple-svc",
            "templateMajorVersion": "1",
            "templateMinorVersion": "0",
            "templateName": "fargate-service"
        }
    ]
}
```

Pour plus d'informations, voir Afficher les données des instances de service dans le Guide de l'administrateur de AWS Proton ou Afficher les données des instances de service dans le Guide de l'utilisateur de AWS Proton.

Exemple 2 : pour répertorier l'instance de service spécifiée

L'`get-service-instance`exemple suivant permet d'obtenir une instance de service.

```
aws proton get-service-instance \
    --name "instance-one" \
    --service-name "simple-svc"
```

Sortie :

```
{
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "deploymentStatus": "SUCCEEDED",
        "environmentName": "simple-env",
        "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
        "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_optional_input: hello world\n
 my_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\n
 environment: my-simple-env\n  spec:\n    my_sample_service_instance_optional_input:
 Ola\n    my_sample_service_instance_required_input: Ciao\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

Pour plus d'informations, voir Afficher les données des instances de service dans le Guide de l'administrateur de AWS Proton ou Afficher les données des instances de service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous ListServiceInstancesà la section Référence des AWS CLI commandes.

`update-service-instance`

L'exemple de code suivant montre comment utiliser`update-service-instance`.

AWS CLI

Pour mettre à jour une instance de service vers une nouvelle version mineure

L'`update-service-instance`exemple suivant met à jour une instance de service vers une nouvelle version secondaire de son modèle de service qui ajoute une nouvelle instance nommée my-other-instance « » avec une nouvelle entrée obligatoire.

```
aws proton update-service-instance \
    --service-name "simple-svc" \
    --spec "file://service-spec.yaml " \
    --template-major-version "1" \
    --template-minor-version "1" \
    --deployment-type "MINOR_VERSION" \
    --name "instance-one"
```

Contenu de `service-spec.yaml` :

```
proton: ServiceSpec
pipeline:
    my_sample_pipeline_optional_input: "abc"
    my_sample_pipeline_required_input: "123"
instances:
    - name: "instance-one"
        environment: "simple-env"
        spec:
            my_sample_service_instance_optional_input: "def"
            my_sample_service_instance_required_input: "456"
    - name: "my-other-instance"
        environment: "simple-env"
        spec:
            my_sample_service_instance_required_input: "789"
```

Sortie :

```
{
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "IN_PROGRESS",
        "environmentName": "arn:aws:proton:region-id:123456789012:environment/
simple-env",
        "lastDeploymentAttemptedAt": "2021-04-02T21:38:00.823000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
```

```
    }
}
```

Pour plus d'informations, voir Mettre à jour une instance de service dans le Guide de l'administrateur de AWS Proton ou Mettre à jour une instance de service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceInstanceà la section Référence des AWS CLI commandes.

`update-service-pipeline`

L'exemple de code suivant montre comment utiliser`update-service-pipeline`.

AWS CLI

Pour mettre à jour un pipeline de services

L'`update-service-pipeline`exemple suivant met à jour un pipeline de services vers une nouvelle version mineure de son modèle de service.

```
aws proton update-service-pipeline \
    --service-name "simple-svc" \
    --spec "file://service-spec.yaml" \
    --template-major-version "1" \
    --template-minor-version "1" \
    --deployment-type "MINOR_VERSION"
```

Sortie :

```
{
    "pipeline": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "IN_PROGRESS",
        "lastDeploymentAttemptedAt": "2021-04-02T21:39:28.991000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
        "spec": "proton: ServiceSpec\n\npipeline:\n
 my_sample_pipeline_optional_input: \"abc\"\n  my_sample_pipeline_required_input:
 \"123\"\n\ninstances:\n  - name: \"my-instance\"\n    environment: \"MySimpleEnv
\"\n    spec:\n      my_sample_service_instance_optional_input: \"def
```

```
\"\n      my_sample_service_instance_required_input: \"456\"\n  - name:
 \"my-other-instance\"\n    environment: \"MySimpleEnv\"\n    spec:\n
 my_sample_service_instance_required_input: \"789\"\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

Pour plus d'informations, voir Mettre à jour un pipeline de services dans le guide de l'administrateur de AWS Proton ou Mettre à jour un pipeline de services dans le guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous UpdateServicePipelineà la section Référence des AWS CLI commandes.

`update-service`

L'exemple de code suivant montre comment utiliser`update-service`.

AWS CLI

Pour mettre à jour un service

L'`update-service`exemple suivant modifie la description d'un service.

```
aws proton update-service \
    --name "MySimpleService" \
    --description "Edit by updating description"
```

Sortie :

```
{
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
        "branchName": "mainline",
        "createdAt": "2021-03-12T22:39:42.318000+00:00",
        "description": "Edit by updating description",
        "lastModifiedAt": "2021-03-12T22:44:21.975000+00:00",
        "name": "MySimpleService",
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
        "repositoryId": "myorg/myapp",
        "status": "ACTIVE",
        "templateName": "fargate-service"
    }
}
```

Pour plus d'informations, voir Modifier un service dans le Guide de l'administrateur de AWS Proton ou Modifier un service dans le Guide de l'utilisateur de AWS Proton.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceà la section Référence des AWS CLI commandes.

Exemples de QLDB utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de QLDB.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-journal-kinesis-stream`

L'exemple de code suivant montre comment utiliser`cancel-journal-kinesis-stream`.

AWS CLI

Pour annuler un flux de journal

L'`cancel-journal-kinesis-stream`exemple suivant annule le flux de journal spécifié à partir d'un registre.

```
aws qldb cancel-journal-kinesis-stream \
```

```
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

Sortie :

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

Pour plus d'informations, consultez la section Diffusion de données de journaux depuis Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CancelJournalKinesisStreamà la section Référence des AWS CLI commandes.

`create-ledger`

L'exemple de code suivant montre comment utiliser`create-ledger`.

AWS CLI

Exemple 1 : pour créer un registre avec des propriétés par défaut

L'`create-ledger`exemple suivant crée un registre avec le nom `myExampleLedger` et le mode d'autorisations. `STANDARD` Les paramètres facultatifs pour la protection contre la suppression et la clé AWS KMS ne sont pas spécifiés. Ils sont donc utilisés par défaut `true` respectivement pour une clé KMS AWS détenue.

```
aws qldb create-ledger \
    --name myExampleLedger \
    --permissions-mode STANDARD
```

Sortie :

```
{
    "State": "CREATING",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": true,
    "CreationDateTime": 1568839243.951,
    "Name": "myExampleLedger",
```

```
    "PermissionsMode": "STANDARD"
}
```

Exemple 2 : pour créer un registre avec la protection contre la suppression désactivée, une clé KMS gérée par le client et des balises spécifiées

L'`create-ledger`exemple suivant crée un registre avec le nom `myExampleLedger2` et le mode d'autorisations. `STANDARD` La fonction de protection contre la suppression est désactivée, la clé KMS gérée par le client spécifiée est utilisée pour le chiffrement au repos et les balises spécifiées sont attachées à la ressource.

```
aws qldb create-ledger \
    --name myExampleLedger2 \
    --permissions-mode STANDARD \
    --no-deletion-protection \
    --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111 \
    --tags IsTest=true,Domain=Test
```

Sortie :

```
{
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger2",
    "DeletionProtection": false,
    "CreationDateTime": 1568839543.557,
    "State": "CREATING",
    "Name": "myExampleLedger2",
    "PermissionsMode": "STANDARD",
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Opérations de base pour Amazon QLDB Ledgers dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous CreateLedgerà la section Référence des AWS CLI commandes.

`delete-ledger`

L'exemple de code suivant montre comment utiliser`delete-ledger`.

AWS CLI

Pour supprimer un registre

L'`delete-ledger`exemple suivant supprime le registre spécifié.

```
aws qldb delete-ledger \
    --name myExampleLedger
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Opérations de base pour Amazon QLDB Ledgers dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous DeleteLedgerà la section Référence des AWS CLI commandes.

`describe-journal-kinesis-stream`

L'exemple de code suivant montre comment utiliser`describe-journal-kinesis-stream`.

AWS CLI

Pour décrire un flux de journal

L'`describe-journal-kinesis-stream`exemple suivant affiche les détails du flux de journal spécifié à partir d'un registre.

```
aws qldb describe-journal-kinesis-stream \
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

Sortie :

```
{
    "Stream": {
        "LedgerName": "myExampleLedger",
        "CreationTime": 1591221984.677,
        "InclusiveStartTime": 1590710400.0,
        "ExclusiveEndTime": 1590796799.0,
        "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",
```

```
        "StreamId": "7ISCkqwe4y25YyHLzYUFAf",
        "Arn": "arn:aws:qldb:us-east-1:123456789012:stream/
myExampleLedger/7ISCkqwe4y25YyHLzYUFAf",
        "Status": "ACTIVE",
        "KinesisConfiguration": {
            "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-
qldb",
            "AggregationEnabled": true
        },
        "StreamName": "myExampleLedger-stream"
    }
}
```

Pour plus d'informations, consultez la section Diffusion de données de journaux depuis Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeJournalKinesisStreamà la section Référence des AWS CLI commandes.

`describe-journal-s3-export`

L'exemple de code suivant montre comment utiliser`describe-journal-s3-export`.

AWS CLI

Pour décrire une tâche d'exportation de journaux

L'`describe-journal-s3-export`exemple suivant affiche les détails de la tâche d'exportation spécifiée à partir d'un registre.

```
aws qldb describe-journal-s3-export \
    --name myExampleLedger \
    --export-id ADR2ONPKN5LINYGb4dp7yZ
```

Sortie :

```
{
    "ExportDescription": {
        "S3ExportConfiguration": {
            "Bucket": "amzn-s3-demo-bucket",
            "Prefix": "ledgerexport1/",
            "EncryptionConfiguration": {
```

```
                "ObjectEncryptionType": "SSE_S3"
            }
        },
        "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
        "Status": "COMPLETED",
        "ExportCreationTime": 1568847801.418,
        "InclusiveStartTime": 1568764800.0,
        "ExclusiveEndTime": 1568847599.0,
        "LedgerName": "myExampleLedger",
        "ExportId": "ADR2ONPKN5LINYGb4dp7yZ"
    }
}
```

Pour plus d'informations, consultez Exporter votre journal dans Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, voir DescribeJournalS3Export dans AWS CLI Command Reference.

`describe-ledger`

L'exemple de code suivant montre comment utiliser`describe-ledger`.

AWS CLI

Pour décrire un registre

L'`describe-ledger`exemple suivant affiche les détails du registre spécifié.

```
aws qldb describe-ledger \
    --name myExampleLedger
```

Sortie :

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "State": "ACTIVE",
    "Name": "myExampleLedger",
    "DeletionProtection": true,
    "PermissionsMode": "STANDARD",
    "EncryptionDescription": {
```

```
        "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "EncryptionStatus": "ENABLED"
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour Amazon QLDB Ledgers dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous DescribeLedgerà la section Référence des AWS CLI commandes.

`export-journal-to-s3`

L'exemple de code suivant montre comment utiliser`export-journal-to-s3`.

AWS CLI

Pour exporter des blocs de journal vers S3

L'`export-journal-to-s3`exemple suivant crée une tâche d'exportation pour des blocs de journaux dans une plage de date et d'heure spécifiée à partir d'un registre portant le nom`myExampleLedger`. La tâche d'exportation écrit les blocs dans un compartiment Amazon S3 spécifié.

```
aws qldb export-journal-to-s3 \
    --name myExampleLedger \
    --inclusive-start-time 2019-09-18T00:00:00Z \
    --exclusive-end-time 2019-09-18T22:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-s3-export-role \
    --s3-export-configuration file://my-s3-export-config.json
```

Contenu de `my-s3-export-config.json` :

```
{
    "Bucket": "amzn-s3-demo-bucket",
    "Prefix": "ledgerexport1/",
    "EncryptionConfiguration": {
        "ObjectEncryptionType": "SSE_S3"
    }
}
```

Sortie :

```
{
    "ExportId": "ADR2ONPKN5LINYGb4dp7yZ"
}
```

Pour plus d'informations, consultez Exporter votre journal dans Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, consultez ExportJournalToS3 dans AWS CLI Command Reference.

`get-block`

L'exemple de code suivant montre comment utiliser`get-block`.

AWS CLI

Exemple 1 : Pour obtenir un bloc de journal et une preuve à des fins de vérification à l'aide de fichiers d'entrée

L'`get-block`exemple suivant demande un objet de données de bloc et une preuve provenant du registre spécifié. La demande concerne une adresse de résumé et une adresse de bloc spécifiées.

```
aws qldb get-block \
    --name vehicle-registration \
    --block-address file://myblockaddress.json \
    --digest-tip-address file://mydigesttipaddress.json
```

Contenu de `myblockaddress.json` :

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"
}
```

Contenu de `mydigesttipaddress.json` :

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
```

```
}
```

Sortie :

```
{
    "Block": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05LOsiKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCBr7pOAIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:
\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\n    VALUE { 'PersonId' :
 'CMVdR77XP8zAglmmFDGTvt' }\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\"VehicleRegistration\",tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\",statements:[0]}}},revisions:[{blockAddress:
{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100},hash:
{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:{VIN:
\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}]}"
    },
    "Proof": {
        "IonText": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT
+qE=}}]"
    }
}
```

Pour plus d'informations, consultez la section Vérification des données dans Amazon QLDB dans le manuel du développeur Amazon QLDB.

Exemple 2 : Pour obtenir un bloc de journal et une preuve à des fins de vérification à l'aide d'une syntaxe abrégée

L'`get-block`exemple suivant demande un objet de données de bloc et une preuve provenant du registre spécifié à l'aide d'une syntaxe abrégée. La demande concerne une adresse de résumé et une adresse de bloc spécifiées.

```
aws qldb get-block \
    --name vehicle-registration \
    --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"'
 \
    --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:123}"'
```

Sortie :

```
{
    "Block": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05LOsiKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCBr7pOAIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:
\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\n    VALUE { 'PersonId' :
 'CMVdR77XP8zAglmmFDGTvt' }\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\"VehicleRegistration\",tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\",statements:[0]}}},revisions:[{blockAddress:
{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100},hash:
{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:{VIN:
\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}]}"
    },
```

```
    "Proof": {
        "IonText": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT
+qE=}}]"
    }
}
```

Pour plus d'informations, consultez la section Vérification des données dans Amazon QLDB dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous GetBlockà la section Référence des AWS CLI commandes.

`get-digest`

L'exemple de code suivant montre comment utiliser`get-digest`.

AWS CLI

Pour obtenir le résumé d'un registre

L'`get-digest`exemple suivant demande un résumé du registre spécifié au dernier bloc validé du journal.

```
aws qldb get-digest \
    --name vehicle-registration
```

Sortie :

```
{
    "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",
    "DigestTipAddress": {
        "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
    }
}
```

Pour plus d'informations, consultez la section Vérification des données dans Amazon QLDB dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous GetDigestà la section Référence des AWS CLI commandes.

`get-revision`

L'exemple de code suivant montre comment utiliser`get-revision`.

AWS CLI

Exemple 1 : Pour obtenir une révision et une preuve de document à des fins de vérification à l'aide de fichiers d'entrée

L'`get-revision`exemple suivant demande un objet de données de révision et une preuve provenant du registre spécifié. La demande concerne une adresse de résumé, un identifiant de document et une adresse de bloc spécifiés pour la révision.

```
aws qldb get-revision \
    --name vehicle-registration \
    --block-address file://myblockaddress.json \
    --document-id JUJgkIcNbhS2goq8RqLuZ4 \
    --digest-tip-address file://mydigesttipaddress.json
```

Contenu de `myblockaddress.json` :

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"
}
```

Contenu de `mydigesttipaddress.json` :

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
}
```

Sortie :

```
{
    "Revision": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:
{VIN:\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
```

```
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}"
    },
    "Proof": {
        "IonText": "[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvBO8Lg+KgpjcnWAU=}},
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},{{l3+EXs69K1+rehlqyWLkt
+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT+qE=}}]"
    }
}
```

Pour plus d'informations, consultez la section Vérification des données dans Amazon QLDB dans le manuel du développeur Amazon QLDB.

Exemple 2 : Pour obtenir une révision de document et une preuve à des fins de vérification à l'aide d'une syntaxe abrégée

L'`get-revision`exemple suivant demande un objet de données de révision et une preuve provenant du registre spécifié à l'aide d'une syntaxe abrégée. La demande concerne une adresse de résumé, un identifiant de document et une adresse de bloc spécifiés pour la révision.

```
aws qldb get-revision \
    --name vehicle-registration \
    --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"'
 \
    --document-id JUJgkIcNbhS2goq8RqLuZ4 \
    --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:123}"'
```

Sortie :

```
{
    "Revision": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:
```

```
{VIN:\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}"
    },
    "Proof": {
        "IonText": "[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvBO8Lg+KgpjcnWAU=}},
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},{{l3+EXs69K1+rehlqyWLkt
+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT+qE=}}]"
    }
}
```

Pour plus d'informations, consultez la section Vérification des données dans Amazon QLDB dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous GetRevisionà la section Référence des AWS CLI commandes.

`list-journal-kinesis-streams-for-ledger`

L'exemple de code suivant montre comment utiliser`list-journal-kinesis-streams-for-ledger`.

AWS CLI

Pour répertorier les flux de journaux d'un registre

L'`list-journal-kinesis-streams-for-ledger`exemple suivant répertorie les flux de journaux pour le registre spécifié.

```
aws qldb list-journal-kinesis-streams-for-ledger \
    --ledger-name myExampleLedger
```

Sortie :

```
{
    "Streams": [
        {
            "LedgerName": "myExampleLedger",
            "CreationTime": 1591221984.677,
            "InclusiveStartTime": 1590710400.0,
            "ExclusiveEndTime": 1590796799.0,
            "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",
            "StreamId": "7ISCkqwe4y25YyHLzYUFAf",
            "Arn": "arn:aws:qldb:us-east-1:123456789012:stream/
myExampleLedger/7ISCkqwe4y25YyHLzYUFAf",
            "Status": "ACTIVE",
            "KinesisConfiguration": {
                "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
for-qldb",
                "AggregationEnabled": true
            },
            "StreamName": "myExampleLedger-stream"
        }
    ]
}
```

Pour plus d'informations, consultez la section Diffusion de données de journaux depuis Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListJournalKinesisStreamsForLedgerà la section Référence des AWS CLI commandes.

`list-journal-s3-exports-for-ledger`

L'exemple de code suivant montre comment utiliser`list-journal-s3-exports-for-ledger`.

AWS CLI

Pour répertorier les tâches d'exportation de journaux pour un registre

L'`list-journal-s3-exports-for-ledger`exemple suivant répertorie les tâches d'exportation de journaux pour le registre spécifié.

```
aws qldb list-journal-s3-exports-for-ledger \
    --name myExampleLedger
```

Sortie :

```
{
    "JournalS3Exports": [
        {
            "LedgerName": "myExampleLedger",
            "ExclusiveEndTime": 1568847599.0,
            "ExportCreationTime": 1568847801.418,
            "S3ExportConfiguration": {
                "Bucket": "amzn-s3-demo-bucket",
                "Prefix": "ledgerexport1/",
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                }
            },
            "ExportId": "ADR2ONPKN5LINYGb4dp7yZ",
            "RoleArn": "arn:aws:iam::123456789012:role/qldb-s3-export",
            "InclusiveStartTime": 1568764800.0,
            "Status": "IN_PROGRESS"
        }
    ]
}
```

Pour plus d'informations, consultez Exporter votre journal dans Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, consultez ListJournalS3 ExportsForLedger dans AWS CLI Command Reference.

`list-journal-s3-exports`

L'exemple de code suivant montre comment utiliser`list-journal-s3-exports`.

AWS CLI

Pour répertorier les offres d'exportation de journaux

L'`list-journal-s3-exports`exemple suivant répertorie les tâches d'exportation de journaux pour tous les livres associés au AWS compte courant et à la région.

```
aws qldb list-journal-s3-exports
```

Sortie :

```
{
    "JournalS3Exports": [
        {
            "Status": "IN_PROGRESS",
            "LedgerName": "myExampleLedger",
            "S3ExportConfiguration": {
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                },
                "Bucket": "amzn-s3-demo-bucket",
                "Prefix": "ledgerexport1/"
            },
            "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
            "ExportCreationTime": 1568847801.418,
            "ExportId": "ADR2ONPKN5LINYGb4dp7yZ",
            "InclusiveStartTime": 1568764800.0,
            "ExclusiveEndTime": 1568847599.0
        },
        {
            "Status": "COMPLETED",
            "LedgerName": "myExampleLedger2",
            "S3ExportConfiguration": {
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                },
                "Bucket": "amzn-s3-demo-bucket",
                "Prefix": "ledgerexport1/"
            },
            "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
            "ExportCreationTime": 1568846847.638,
            "ExportId": "2pdvW8UQrjBAiYTMehEJDI",
            "InclusiveStartTime": 1568592000.0,
            "ExclusiveEndTime": 1568764800.0
        }
    ]
}
```

Pour plus d'informations, consultez Exporter votre journal dans Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, consultez ListJournalS3Exports dans AWS CLI Command Reference.

`list-ledgers`

L'exemple de code suivant montre comment utiliser`list-ledgers`.

AWS CLI

Pour répertorier vos registres disponibles

L'`list-ledgers`exemple suivant répertorie tous les registres associés au AWS compte courant et à la région.

```
aws qldb list-ledgers
```

Sortie :

```
{
    "Ledgers": [
        {
            "State": "ACTIVE",
            "CreationDateTime": 1568839243.951,
            "Name": "myExampleLedger"
        },
        {
            "State": "ACTIVE",
            "CreationDateTime": 1568839543.557,
            "Name": "myExampleLedger2"
        }
    ]
}
```

Pour plus d'informations, consultez la section Opérations de base pour Amazon QLDB Ledgers dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, reportez-vous ListLedgersà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises associées à un registre

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises associées au registre spécifié.

```
aws qldb list-tags-for-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger
```

Sortie :

```
{
    "Tags": {
        "IsTest": "true",
        "Domain": "Test"
    }
}
```

Pour plus d'informations, consultez la section Marquage des ressources Amazon QLDB dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`stream-journal-to-kinesis`

L'exemple de code suivant montre comment utiliser`stream-journal-to-kinesis`.

AWS CLI

Exemple 1 : pour diffuser des données de journal vers Kinesis Data Streams à l'aide de fichiers d'entrée

L'`stream-journal-to-kinesis`exemple suivant crée un flux de données de journal dans une plage de dates et d'heures spécifiée à partir d'un registre portant le nom`myExampleLedger`. Le flux envoie les données à un flux de données Amazon Kinesis spécifié.

```
aws qldb stream-journal-to-kinesis \
    --ledger-name myExampleLedger \
    --inclusive-start-time 2020-05-29T00:00:00Z \
    --exclusive-end-time 2020-05-29T23:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
    --kinesis-configuration file://my-kinesis-config.json \
    --stream-name myExampleLedger-stream
```

Contenu de `my-kinesis-config.json` :

```
{
    "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",
    "AggregationEnabled": true
}
```

Sortie :

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

Pour plus d'informations, consultez la section Diffusion de données de journaux depuis Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

Exemple 2 : pour diffuser des données de journal vers Kinesis Data Streams à l'aide d'une syntaxe abrégée

L'`stream-journal-to-kinesis`exemple suivant crée un flux de données de journal dans une plage de dates et d'heures spécifiée à partir d'un registre portant le nom`myExampleLedger`. Le flux envoie les données à un flux de données Amazon Kinesis spécifié.

```
aws qldb stream-journal-to-kinesis \
    --ledger-name myExampleLedger \
    --inclusive-start-time 2020-05-29T00:00:00Z \
    --exclusive-end-time 2020-05-29T23:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
    --stream-name myExampleLedger-stream \
    --kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/
stream-for-qldb,AggregationEnabled=true
```

Sortie :

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

Pour plus d'informations, consultez la section Diffusion de données de journaux depuis Amazon QLDB dans le manuel Amazon QLDB Developer Guide.

- Pour plus de détails sur l'API, voir StreamJournalToKinesisla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour étiqueter un registre

L'`tag-resource`exemple suivant ajoute un ensemble de balises à un registre spécifié.

```
aws qldb tag-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \
    --tags IsTest=true,Domain=Test
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon QLDB dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`untag-resource`exemple suivant supprime les balises avec les clés de balise spécifiées d'un registre spécifié.

```
aws qldb untag-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \
    --tag-keys IsTest Domain
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon QLDB dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-ledger-permissions-mode`

L'exemple de code suivant montre comment utiliser`update-ledger-permissions-mode`.

AWS CLI

Exemple 1 : pour mettre à jour le mode d'autorisations d'un registre sur STANDARD

L'`update-ledger-permissions-mode`exemple suivant attribue le mode `STANDARD` d'autorisations au registre spécifié.

```
aws qldb update-ledger-permissions-mode \
    --name myExampleLedger \
    --permissions-mode STANDARD
```

Sortie :

```
{
    "Name": "myExampleLedger",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "PermissionsMode": "STANDARD"
}
```

Exemple 2 : pour mettre à jour le mode d'autorisations d'un registre sur ALLOW_ALL

L'`update-ledger-permissions-mode`exemple suivant attribue le mode `ALLOW_ALL` d'autorisations au registre spécifié.

```
aws qldb update-ledger-permissions-mode \
```

```
    --name myExampleLedger \
    --permissions-mode ALLOW_ALL
```

Sortie :

```
{
    "Name": "myExampleLedger",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "PermissionsMode": "ALLOW_ALL"
}
```

Pour plus d'informations, consultez la section Opérations de base pour Amazon QLDB Ledgers dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, voir UpdateLedgerPermissionsModela section Référence des AWS CLI commandes.

`update-ledger`

L'exemple de code suivant montre comment utiliser`update-ledger`.

AWS CLI

Exemple 1 : pour mettre à jour la propriété de protection contre la suppression d'un registre

L'`update-ledger`exemple suivant met à jour le registre spécifié pour désactiver la fonctionnalité de protection contre la suppression.

```
aws qldb update-ledger \
    --name myExampleLedger \
    --no-deletion-protection
```

Sortie :

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE"
```

```
}
```

Exemple 2 : pour mettre à jour la clé AWS KMS d'un registre en une clé gérée par le client

L'`update-ledger`exemple suivant met à jour le registre spécifié afin d'utiliser une clé KMS gérée par le client pour le chiffrement au repos.

```
aws qldb update-ledger \
    --name myExampleLedger \
    --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111
```

Sortie :

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE",
    "EncryptionDescription": {
        "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "EncryptionStatus": "UPDATING"
    }
}
```

Exemple 3 : pour mettre à jour la clé AWS KMS d'un registre en une clé AWS détenue

L'`update-ledger`exemple suivant met à jour le registre spécifié afin d'utiliser une clé KMS AWS détenue pour le chiffrement au repos.

```
aws qldb update-ledger \
    --name myExampleLedger \
    --kms-key AWS_OWNED_KMS_KEY
```

Sortie :

```
{
    "CreationDateTime": 1568839243.951,
```

```
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE",
    "EncryptionDescription": {
        "KmsKeyArn": "AWS_OWNED_KMS_KEY",
        "EncryptionStatus": "UPDATING"
    }
}
```

Pour plus d'informations, consultez la section Opérations de base pour Amazon QLDB Ledgers dans le manuel du développeur Amazon QLDB.

- Pour plus de détails sur l'API, voir UpdateLedgerla section Référence des AWS CLI commandes.

Exemples d'Amazon RDS utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon RDS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-option-to-option-group`

L'exemple de code suivant montre comment utiliser`add-option-to-option-group`.

AWS CLI

Pour ajouter une option à un groupe d'options

L'`add-option-to-option-group`exemple suivant ajoute une option au groupe d'options spécifié.

```
aws rds add-option-to-option-group \
    --option-group-name myoptiongroup \
    --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \
    --apply-immediately
```

Sortie :

```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Test Option Group",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "12.1",
        "Options": [
            {
                "OptionName": "Timezone",
                "OptionDescription": "Change time zone",
                "Persistent": true,
                "Permanent": false,
                "OptionSettings": [
                    {
                        "Name": "TIME_ZONE",
                        "Value": "Australia/Sydney",
                        "DefaultValue": "UTC",
                        "Description": "Specifies the timezone the user wants to
 change the system time to",
                        "ApplyType": "DYNAMIC",
                        "DataType": "STRING",
                        "AllowedValues": "Africa/Cairo,Africa/Casablanca,Africa/
Harare,Africa/Lagos,Africa/Luanda,Africa/Monrovia,Africa/Nairobi,Africa/
Tripoli,Africa/Windhoek,America/Araguaina,America/Argentina/Buenos_Aires,America/
Asuncion,America/Bogota,America/Caracas,America/Chicago,America/Chihuahua,America/
Cuiaba,America/Denver,America/Detroit,America/Fortaleza,America/Godthab,America/
Guatemala,America/Halifax,America/Lima,America/Los_Angeles,America/Manaus,America/
Matamoros,America/Mexico_City,America/Monterrey,America/Montevideo,America/
New_York,America/Phoenix,America/Santiago,America/Sao_Paulo,America/Tijuana,America/
Toronto,Asia/Amman,Asia/Ashgabat,Asia/Baghdad,Asia/Baku,Asia/Bangkok,Asia/
Beirut,Asia/Calcutta,Asia/Damascus,Asia/Dhaka,Asia/Hong_Kong,Asia/Irkutsk,Asia/
Jakarta,Asia/Jerusalem,Asia/Kabul,Asia/Karachi,Asia/Kathmandu,Asia/Kolkata,Asia/
Krasnoyarsk,Asia/Magadan,Asia/Manila,Asia/Muscat,Asia/Novosibirsk,Asia/Rangoon,Asia/
```

```
Riyadh,Asia/Seoul,Asia/Shanghai,Asia/Singapore,Asia/Taipei,Asia/Tehran,Asia/
Tokyo,Asia/Ulaanbaatar,Asia/Vladivostok,Asia/Yakutsk,Asia/Yerevan,Atlantic/
Azores,Atlantic/Cape_Verde,Australia/Adelaide,Australia/Brisbane,Australia/
Darwin,Australia/Eucla,Australia/Hobart,Australia/Lord_Howe,Australia/
Perth,Australia/Sydney,Brazil/DeNoronha,Brazil/East,Canada/Newfoundland,Canada/
Saskatchewan,Etc/GMT-3,Europe/Amsterdam,Europe/Athens,Europe/Berlin,Europe/
Dublin,Europe/Helsinki,Europe/Kaliningrad,Europe/London,Europe/Madrid,Europe/
Moscow,Europe/Paris,Europe/Prague,Europe/Rome,Europe/Sarajevo,Pacific/Apia,Pacific/
Auckland,Pacific/Chatham,Pacific/Fiji,Pacific/Guam,Pacific/Honolulu,Pacific/
Kiritimati,Pacific/Marquesas,Pacific/Samoa,Pacific/Tongatapu,Pacific/Wake,US/
Alaska,US/Central,US/East-Indiana,US/Eastern,US/Pacific,UTC",
                        "IsModifiable": true,
                        "IsCollection": false
                    }
                ],
                "DBSecurityGroupMemberships": [],
                "VpcSecurityGroupMemberships": []
            },
            {
                "OptionName": "OEM",
                "OptionDescription": "Oracle 12c EM Express",
                "Persistent": false,
                "Permanent": false,
                "Port": 5500,
                "OptionSettings": [],
                "DBSecurityGroupMemberships": [
                    {
                        "DBSecurityGroupName": "default",
                        "Status": "authorized"
                    }
                ],
                "VpcSecurityGroupMemberships": []
            }
        ],
        "AllowsVpcAndNonVpcInstanceMemberships": false,
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
    }
}
```

Pour plus d'informations, consultez Ajouter une option à un groupe d'options dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir AddOptionToOptionGroupla section Référence des AWS CLI commandes.

`add-role-to-db-cluster`

L'exemple de code suivant montre comment utiliser`add-role-to-db-cluster`.

AWS CLI

Pour associer un rôle AWS Identity and Access Management (IAM) à un cluster de bases de données

L'`add-role-to-db-cluster`exemple suivant associe un rôle à un cluster de bases de données.

```
aws rds add-role-to-db-cluster \
    --db-cluster-identifier mydbcluster \
    --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Associer un rôle IAM à un cluster de bases de données Amazon Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir AddRoleToDbClusterla section Référence des AWS CLI commandes.

`add-role-to-db-instance`

L'exemple de code suivant montre comment utiliser`add-role-to-db-instance`.

AWS CLI

Pour associer un rôle AWS Identity and Access Management (IAM) à une instance de base de données

L'`add-role-to-db-instance`exemple suivant ajoute le rôle à une instance de base de données Oracle nommée`test-instance`.

```
aws rds add-role-to-db-instance \
    --db-instance-identifier test-instance \
    --feature-name S3_INTEGRATION \
    --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Conditions requises pour l'intégration d'Amazon RDS Oracle à Amazon S3 dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir AddRoleToDbInstancela section Référence des AWS CLI commandes.

`add-source-identifier-to-subscription`

L'exemple de code suivant montre comment utiliser`add-source-identifier-to-subscription`.

AWS CLI

Pour ajouter un identifiant de source à un abonnement

L'`add-source-identifier`exemple suivant ajoute un autre identifiant de source à un abonnement existant.

```
aws rds add-source-identifier-to-subscription \
    --subscription-name my-instance-events \
    --source-identifier test-instance-repl
```

Sortie :

```
{
    "EventSubscription": {
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "CustSubscriptionId": "my-instance-events",
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "Enabled": false,
        "Status": "modifying",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "CustomerAwsId": "123456789012",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "SourceType": "db-instance",
        "SourceIdsList": [
            "test-instance",
            "test-instance-repl"
```

```
        ]
    }
}
```

- Pour plus de détails sur l'API, voir AddSourceIdentifierToSubscriptionla section Référence des AWS CLI commandes.

`add-tags-to-resource`

L'exemple de code suivant montre comment utiliser`add-tags-to-resource`.

AWS CLI

Pour ajouter des balises à une ressource

L'`add-tags-to-resource`exemple suivant ajoute des balises à une base de données RDS.

```
aws rds add-tags-to-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:database-mysql \
    --tags "[{\"Key\": \"Name\",\"Value\": \"MyDatabase\"},{\"Key\": \"Environment
\",\"Value\": \"test\"}]"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon RDS dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir AddTagsToResourcela section Référence des AWS CLI commandes.

`apply-pending-maintenance-action`

L'exemple de code suivant montre comment utiliser`apply-pending-maintenance-action`.

AWS CLI

Pour appliquer les actions de maintenance en attente

L'`apply-pending-maintenance-action`exemple suivant applique les actions de maintenance en attente pour un cluster de base de données.

```
aws rds apply-pending-maintenance-action \
```

```
    --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:my-db-cluster \
    --apply-action system-update \
    --opt-in-type immediate
```

Sortie :

```
{
    "ResourcePendingMaintenanceActions": {
        "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:cluster:my-db-
cluster",
        "PendingMaintenanceActionDetails": [
            {
                "Action": "system-update",
                "OptInStatus": "immediate",
                "CurrentApplyDate": "2021-01-23T01:07:36.100Z",
                "Description": "Upgrade to Aurora PostgreSQL 3.3.2"
            }
        ]
    }
}
```

Pour plus d'informations, consultez Maintenance d'une instance de base de données dans le guide de l'utilisateur Amazon RDS et Maintenance d'un cluster de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir ApplyPendingMaintenanceActionla section Référence des AWS CLI commandes.

`authorize-db-security-group-ingress`

L'exemple de code suivant montre comment utiliser`authorize-db-security-group-ingress`.

AWS CLI

Pour associer un rôle AWS Identity and Access Management (IAM) à une instance de base de données

L'`authorize-db-security-group-ingress`exemple suivant configure le groupe de sécurité par défaut avec une règle d'entrée pour la plage d'adresses IP CIDR 192.0.2.0/24.

```
aws rds authorize-db-security-group-ingress \
    --db-security-group-name default \
```

```
    --cidrip 192.0.2.0/24
```

Sortie :

```
{
    "DBSecurityGroup": {
        "OwnerId": "123456789012",
        "DBSecurityGroupName": "default",
        "DBSecurityGroupDescription": "default",
        "EC2SecurityGroups": [],
        "IPRanges": [
            {
                "Status": "authorizing",
                "CIDRIP": "192.0.2.0/24"
            }
        ],
        "DBSecurityGroupArn": "arn:aws:rds:us-east-1:111122223333:secgrp:default"
    }
}
```

Pour plus d'informations, consultez Autoriser l'accès réseau à un groupe de sécurité de base de données à partir d'une plage d'adresses IP dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir AuthorizeDbSecurityGroupIngressla section Référence des AWS CLI commandes.

`backtrack-db-cluster`

L'exemple de code suivant montre comment utiliser`backtrack-db-cluster`.

AWS CLI

Pour revenir en arrière sur un cluster de base de données Aurora

L'`backtrack-db-cluster`exemple suivant revient sur l'exemple de cluster de base de données spécifié jusqu'au 19 mars 2018 à 10 heures.

```
aws rds backtrack-db-cluster --db-cluster-identifier sample-cluster --backtrack-
to 2018-03-19T10:00:00+00:00
```

Cette commande génère un bloc JSON qui accuse réception de la modification apportée à la ressource RDS.

- Pour plus de détails sur l'API, voir BacktrackDbClusterla section Référence des AWS CLI commandes.

`cancel-export-task`

L'exemple de code suivant montre comment utiliser`cancel-export-task`.

AWS CLI

Pour annuler l'exportation d'un instantané vers Amazon S3

L'`cancel-export-task`exemple suivant annule une tâche d'exportation en cours qui exporte un instantané vers Amazon S3.

```
aws rds cancel-export-task \
    --export-task-identifier my-s3-export-1
```

Sortie :

```
{
    "ExportTaskIdentifier": "my-s3-export-1",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:publisher-final-
snapshot",
    "SnapshotTime": "2019-03-24T20:01:09.815Z",
    "S3Bucket": "amzn-s3-demo-bucket",
    "S3Prefix": "",
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/export-snap-S3-role",
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcd0000-7bfd-4594-af38-
aabbccddeeff",
    "Status": "CANCELING",
    "PercentProgress": 0,
    "TotalExtractedDataInGB": 0
}
```

Pour plus d'informations, consultez Annulation d'une tâche d'exportation d'instantanés dans le guide de l'utilisateur Amazon RDS ou Annulation d'une tâche d'exportation d'instantanés dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CancelExportTaskla section Référence des AWS CLI commandes.

`copy-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`copy-db-cluster-parameter-group`.

AWS CLI

Pour copier un groupe de paramètres de cluster de base de données

L'`copy-db-cluster-parameter-group`exemple suivant crée une copie d'un groupe de paramètres de cluster de base de données.

```
aws rds copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier mydbclusterpg \
    --target-db-cluster-parameter-group-identifier mydbclusterpgcopy \
    --target-db-cluster-parameter-group-description "Copy of mydbclusterpg parameter
 group"
```

Sortie :

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterpgcopy",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterpgcopy",
        "DBParameterGroupFamily": "aurora-mysql5.7",
        "Description": "Copy of mydbclusterpg parameter group"
    }
}
```

Pour plus d'informations, consultez Copier un groupe de paramètres de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CopyDbClusterParameterGroupla section Référence des AWS CLI commandes.

`copy-db-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`copy-db-cluster-snapshot`.

AWS CLI

Pour copier un instantané d'un cluster de base de données

L'`copy-db-cluster-snapshot`exemple suivant crée une copie d'un instantané de cluster de base de données, y compris ses balises.

```
aws rds copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:rds:myaurora-2019-06-04-09-16
    --target-db-cluster-snapshot-identifier myclustersnapshotcopy \
    --copy-tags
```

Sortie :

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
        "DBClusterIdentifier": "myaurora",
        "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 0,
        "Status": "available",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
        "SnapshotType": "manual",
        "PercentProgress": 100,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:myclustersnapshotcopy",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

Pour plus d'informations, consultez Copier un instantané dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CopyDbClusterSnapshotla section Référence des AWS CLI commandes.

`copy-db-parameter-group`

L'exemple de code suivant montre comment utiliser`copy-db-parameter-group`.

AWS CLI

Pour copier un groupe de paramètres de cluster de base de données

L'`copy-db-parameter-group`exemple suivant crée une copie d'un groupe de paramètres de base de données.

```
aws rds copy-db-parameter-group \
    --source-db-parameter-group-identifier mydbpg \
    --target-db-parameter-group-identifier mydbpgcopy \
    --target-db-parameter-group-description "Copy of mydbpg parameter group"
```

Sortie :

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbpgcopy",
        "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",
        "DBParameterGroupFamily": "mysql5.7",
        "Description": "Copy of mydbpg parameter group"
    }
}
```

Pour plus d'informations, consultez Copier un groupe de paramètres de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir CopyDbParameterGroupla section Référence des AWS CLI commandes.

`copy-db-snapshot`

L'exemple de code suivant montre comment utiliser`copy-db-snapshot`.

AWS CLI

Pour copier un instantané de base de données

L'`copy-db-snapshot`exemple suivant crée une copie d'un instantané de base de données.

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier rds:database-mysql-2019-06-06-08-38
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Sortie :

```
{
    "DBSnapshot": {
        "VpcId": "vpc-6594f31c",
        "Status": "creating",
        "Encrypted": true,
        "SourceDBSnapshotIdentifier": "arn:aws:rds:us-
east-1:123456789012:snapshot:rds:database-mysql-2019-06-06-08-38",
        "MasterUsername": "admin",
        "Iops": 1000,
        "Port": 3306,
        "LicenseModel": "general-public-license",
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshotcopy",
        "EngineVersion": "5.6.40",
        "OptionGroupName": "default:mysql-5-6",
        "ProcessorFeatures": [],
        "Engine": "mysql",
        "StorageType": "io1",
        "DbiResourceId": "db-ZI7UJ5BLKMBYFGX7FDENCKADC4",
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "SnapshotType": "manual",
        "IAMDatabaseAuthenticationEnabled": false,
        "SourceRegion": "us-east-1",
        "DBInstanceIdentifier": "database-mysql",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "AvailabilityZone": "us-east-1f",
        "PercentProgress": 0,
        "AllocatedStorage": 100,
        "DBSnapshotIdentifier": "mydbsnapshotcopy"
    }
}
```

Pour plus d'informations, consultez Copier un instantané dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir CopyDbSnapshotla section Référence des AWS CLI commandes.

`copy-option-group`

L'exemple de code suivant montre comment utiliser`copy-option-group`.

AWS CLI

Pour copier un groupe d'options

L'`copy-option-group`exemple suivant crée une copie d'un groupe d'options.

```
aws rds copy-option-group \
    --source-option-group-identifier myoptiongroup \
    --target-option-group-identifier new-option-group \
    --target-option-group-description "My option group copy"
```

Sortie :

```
{
    "OptionGroup": {
        "Options": [],
        "OptionGroupName": "new-option-group",
        "MajorEngineVersion": "11.2",
        "OptionGroupDescription": "My option group copy",
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "EngineName": "oracle-ee",
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:new-option-group"
    }
}
```

Pour plus d'informations, consultez la section Création d'une copie d'un groupe d'options dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir CopyOptionGroupla section Référence des AWS CLI commandes.

`create-blue-green-deployment`

L'exemple de code suivant montre comment utiliser`create-blue-green-deployment`.

AWS CLI

Exemple 1 : pour créer un blue/green déploiement pour une instance de base de données RDS pour MySQL

L'`create-blue-green-deployment`exemple suivant crée un blue/green déploiement pour une instance de base de données MySQL.

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name bgd-cli-test-instance \
    --source arn:aws:rds:us-east-1:123456789012:db:my-db-instance \
    --target-engine-version 8.0 \
    --target-db-parameter-group-name mysql-80-group
```

Sortie :

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
        "BlueGreenDeploymentName": "bgd-cli-test-instance",
        "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3"
            }
        ],
```

```
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "PENDING"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "PENDING"
            },
            {
                "Name": "CONFIGURE_BACKUPS",
                "Status": "PENDING"
            },
            {
                "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                "Status": "PENDING"
            }
        ],
        "Status": "PROVISIONING",
        "CreateTime": "2022-02-25T21:18:51.183000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Création d'un blue/green déploiement dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour créer un blue/green déploiement pour un cluster de base de données Aurora MySQL

L'`create-blue-green-deployment`exemple suivant crée un blue/green déploiement pour un cluster de base de données Aurora MySQL.

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name my-blue-green-deployment \
    --source arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster \
    --target-engine-version 8.0 \
    --target-db-cluster-parameter-group-name ams-80-binlog-enabled \
    --target-db-parameter-group-name mysql-80-cluster-group
```

Sortie :

```
{
      "BlueGreenDeployment": {
```

```
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "SwitchoverDetails": [
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-
mysql-cluster",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-1",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-2",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-3",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
excluded-member-endpoint",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
reader-endpoint",
            "Status": "PROVISIONING"
          }
        ],
        "Tasks": [
          {
            "Name": "CREATING_READ_REPLICA_OF_SOURCE",
            "Status": "PENDING"
          },
          {
            "Name": "DB_ENGINE_VERSION_UPGRADE",
            "Status": "PENDING"
```

```
            },
            {
              "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
              "Status": "PENDING"
            },
            {
              "Name": "CREATE_CUSTOM_ENDPOINTS",
              "Status": "PENDING"
            }
          ],
          "Status": "PROVISIONING",
          "CreateTime": "2022-02-25T21:12:00.288000+00:00"
        }
}
```

Pour plus d'informations, consultez la section Création d'un blue/green déploiement dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateBlueGreenDeploymentla section Référence des AWS CLI commandes.

`create-db-cluster-endpoint`

L'exemple de code suivant montre comment utiliser`create-db-cluster-endpoint`.

AWS CLI

Pour créer un point de terminaison de cluster de base de données personnalisé

L'`create-db-cluster-endpoint`exemple suivant crée un point de terminaison de cluster de base de données personnalisé et l'associe au cluster de base de données Aurora spécifié.

```
aws rds create-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint \
    --endpoint-type reader \
    --db-cluster-identifier mydbcluster \
    --static-members dbinstance1 dbinstance2
```

Sortie :

```
{
```

```
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "creating",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

Pour plus d'informations, consultez la section Gestion des connexions Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbClusterEndpointla section Référence des AWS CLI commandes.

`create-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`create-db-cluster-parameter-group`.

AWS CLI

Pour créer un groupe de paramètres de cluster de base de données

L'`create-db-cluster-parameter-group`exemple suivant crée un groupe de paramètres de cluster de base de données.

```
aws rds create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --db-parameter-group-family aurora5.6 \
    --description "My new cluster parameter group"
```

Sortie :

```
{
```

```
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "aurora5.6",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de paramètres de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbClusterParameterGroupla section Référence des AWS CLI commandes.

`create-db-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`create-db-cluster-snapshot`.

AWS CLI

Pour créer un instantané de cluster de base de données

L'`create-db-cluster-snapshot`exemple suivant crée un instantané de cluster de base de données.

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mydbcluster \
    --db-cluster-snapshot-identifier mydbclustersnapshot
```

Sortie :

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "DBClusterSnapshotIdentifier": "mydbclustersnapshot",
        "DBClusterIdentifier": "mydbcluster",
```

```
        "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 1,
        "Status": "creating",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:mydbclustersnapshot",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

Pour plus d'informations, consultez la section Création d'un instantané de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbClusterSnapshotla section Référence des AWS CLI commandes.

`create-db-cluster`

L'exemple de code suivant montre comment utiliser`create-db-cluster`.

AWS CLI

Exemple 1 : pour créer un cluster de base de données compatible avec MySQL 5.7

L'`create-db-cluster`exemple suivant crée un cluster de base de données compatible MySQL 5.7 en utilisant la version du moteur par défaut. Remplacez le mot de passe d'exemple `secret99` par un mot de passe sécurisé. Lorsque vous utilisez la console pour créer un cluster de base de données, Amazon RDS crée automatiquement l'instance de base de données d'écriture pour votre cluster de base de données. Toutefois, lorsque vous utilisez l' AWS interface de ligne de commande pour créer un cluster de base de données, vous devez créer explicitement l'instance de base de données d'écriture pour votre cluster de base de données à l'aide de la commande `create-db-instance` AWS CLI.

```
aws rds create-db-cluster \
    --db-cluster-identifier sample-cluster \
    --engine aurora-mysql \
    --engine-version 5.7 \
    --master-username admin \
    --master-user-password secret99 \
    --db-subnet-group-name default \
    --vpc-security-group-ids sg-0b9130572daf3dc16
```

Sortie :


```
{
    "DBCluster": {
        "DBSubnetGroup": "default",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
                "Status": "active"
            }
        ],
        "AllocatedStorage": 1,
        "AssociatedRoles": [],
        "PreferredBackupWindow": "09:12-09:42",
        "ClusterCreateTime": "2023-02-27T23:21:33.048Z",
        "DeletionProtection": false,
        "IAMDatabaseAuthenticationEnabled": false,
        "ReadReplicaIdentifiers": [],
        "EngineMode": "provisioned",
        "Engine": "aurora-mysql",
        "StorageEncrypted": false,
        "MultiAZ": false,
        "PreferredMaintenanceWindow": "mon:04:31-mon:05:01",
        "HttpEndpointEnabled": false,
        "BackupRetentionPeriod": 1,
        "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "MasterUsername": "master",
        "EngineVersion": "5.7.mysql_aurora.2.11.1",
```

```
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "DBClusterMembers": [],
        "Port": 3306,
        "Status": "creating",
        "Endpoint": "sample-cluster.cluster-cnpexample.us-east-1.rds.amazonaws.com",
        "DBClusterParameterGroup": "default.aurora-mysql5.7",
        "HostedZoneId": "Z2R2ITUGPM61AM",
        "ReaderEndpoint": "sample-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "CopyTagsToSnapshot": false
    }
}
```

Exemple 2 : pour créer un cluster de base de données compatible avec PostgreSQL

L'`create-db-cluster`exemple suivant crée un cluster de base de données compatible avec PostgreSQL en utilisant la version du moteur par défaut. Remplacez le mot de passe d'exemple `secret99` par un mot de passe sécurisé. Lorsque vous utilisez la console pour créer un cluster de base de données, Amazon RDS crée automatiquement l'instance de base de données d'écriture pour votre cluster de base de données. Toutefois, lorsque vous utilisez l' AWS interface de ligne de commande pour créer un cluster de base de données, vous devez créer explicitement l'instance de base de données d'écriture pour votre cluster de base de données à l'aide de la commande `create-db-instance` AWS CLI.

```
aws rds create-db-cluster \
    --db-cluster-identifier sample-pg-cluster \
    --engine aurora-postgresql \
    --master-username master \
    --master-user-password secret99 \
    --db-subnet-group-name default \
    --vpc-security-group-ids sg-0b9130572daf3dc16
```

Sortie :

```
{
    "DBCluster": {
        "Endpoint": "sample-pg-cluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
        "HttpEndpointEnabled": false,
        "DBClusterMembers": [],
        "EngineMode": "provisioned",
        "CopyTagsToSnapshot": false,
```

```
        "HostedZoneId": "Z2R2ITUGPM61AM",
        "IAMDatabaseAuthenticationEnabled": false,
        "AllocatedStorage": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
                "Status": "active"
            }
        ],
        "DeletionProtection": false,
        "StorageEncrypted": false,
        "BackupRetentionPeriod": 1,
        "PreferredBackupWindow": "09:56-10:26",
        "ClusterCreateTime": "2023-02-27T23:26:08.371Z",
        "DBClusterParameterGroup": "default.aurora-postgresql13",
        "EngineVersion": "13.7",
        "Engine": "aurora-postgresql",
        "Status": "creating",
        "DBClusterIdentifier": "sample-pg-cluster",
        "MultiAZ": false,
        "Port": 5432,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-pg-
cluster",
        "AssociatedRoles": [],
        "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
        "PreferredMaintenanceWindow": "wed:03:33-wed:04:03",
        "ReaderEndpoint": "sample-pg-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "MasterUsername": "master",
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "ReadReplicaIdentifiers": [],
        "DBSubnetGroup": "default"
    }
}
```

Pour plus d'informations, consultez la section Création d'un cluster de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbClusterla section Référence des AWS CLI commandes.

`create-db-instance-read-replica`

L'exemple de code suivant montre comment utiliser`create-db-instance-read-replica`.

AWS CLI

Pour créer une instance de base de données, lisez la réplique

Cet exemple crée une réplique en lecture d'une instance de base de données existante nommée`test-instance`. La réplique lue est nommée`test-instance-repl`.

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier test-instance-repl \
    --source-db-instance-identifier test-instance
```

Sortie :

```
{
    "DBInstance": {
        "IAMDatabaseAuthenticationEnabled": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
        "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
        "DBInstanceIdentifier": "test-instance-repl",
        ...some output truncated...
    }
}
```

- Pour plus de détails sur l'API, voir CreateDbInstanceReadReplicala section Référence des AWS CLI commandes.

`create-db-instance`

L'exemple de code suivant montre comment utiliser`create-db-instance`.

AWS CLI

Pour créer une instance de base de données

L'`create-db-instance`exemple suivant utilise les options requises pour lancer une nouvelle instance de base de données.

```
aws rds create-db-instance \
    --db-instance-identifier test-mysql-instance \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --master-username admin \
    --master-user-password secret99 \
    --allocated-storage 20
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
```

```
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-########",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-########",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-########",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ]
    },
    "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
    "PendingModifiedValues": {
        "MasterUserPassword": "****"
    },
    "MultiAZ": false,
    "EngineVersion": "5.7.22",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",
    "OptionGroupMemberships": [
        {
            "OptionGroupName": "default:mysql-5-7",
            "Status": "in-sync"
        }
    ],
    "PubliclyAccessible": true,
    "StorageType": "gp2",
    "DbInstancePort": 0,
```

```
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

Pour plus d'informations, consultez la section Création d'une instance de base de données Amazon RDS dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Create DBInstance in AWS CLI Command Reference.

`create-db-parameter-group`

L'exemple de code suivant montre comment utiliser`create-db-parameter-group`.

AWS CLI

Pour créer un groupe de paramètres de base de données

L'`create-db-parameter-group`exemple suivant crée un groupe de paramètres de base de données.

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
    --description "My new parameter group"
```

Sortie :

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
```

```
        "Description": "My new parameter group",
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

Pour plus d'informations, consultez la section Création d'un groupe de paramètres de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Créer un DBParameter groupe dans AWS CLI la référence des commandes.

`create-db-proxy-endpoint`

L'exemple de code suivant montre comment utiliser`create-db-proxy-endpoint`.

AWS CLI

Pour créer un point de terminaison proxy de base de données pour une base de données RDS

L'`create-db-proxy-endpoint`exemple suivant crée un point de terminaison de proxy de base de données.

```
aws rds create-db-proxy-endpoint \
    --db-proxy-name proxyExample \
    --db-proxy-endpoint-name "proxyep1" \
    --vpc-subnet-ids subnetgroup1 subnetgroup2
```

Sortie :

```
{
"DBProxyEndpoint": {
        "DBProxyEndpointName": "proxyep1",
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
        "DBProxyName": "proxyExample",
        "Status": "creating",
        "VpcId": "vpc-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234",
            "sg-5678"
        ],
```

```
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyep1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": false
    }
}
```

Pour plus d'informations, consultez Création d'un point de terminaison proxy dans le guide de l'utilisateur Amazon RDS et Création d'un point de terminaison proxy dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbProxyEndpointla section Référence des AWS CLI commandes.

`create-db-proxy`

L'exemple de code suivant montre comment utiliser`create-db-proxy`.

AWS CLI

Pour créer un proxy de base de données pour une base de données RDS

L'`create-db-proxy`exemple suivant crée un proxy de base de données.

```
aws rds create-db-proxy \
    --db-proxy-name proxyExample \
    --engine-family MYSQL \
    --auth
 Description="proxydescription1",AuthScheme="SECRETS",SecretArn="arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",IAMAuth="DISABLED",ClientPasswordAuthType="MYSQ
 \
    --role-arn arn:aws:iam::123456789123:role/ProxyRole \
    --vpc-subnet-ids subnetgroup1 subnetgroup2
```

Sortie :

```
{
```

```
"DBProxy": {
        "DBProxyName": "proxyExample",
        "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
        "EngineFamily": "MYSQL",
        "VpcId": "vpc-1234567",
        "VpcSecuritytGroupIds": [
            "sg-1234",
            "sg-5678",
            "sg-9101"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Auth": "[
            {
                "Description": "proxydescription1",
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
            }
        ]",
        "RoleArn": "arn:aws:iam::12345678912:role/ProxyRole",
        "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
        "RequireTLS": false,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

Pour plus d'informations, consultez Création d'un proxy RDS dans le guide de l'utilisateur Amazon RDS et Création d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbProxyla section Référence des AWS CLI commandes.

`create-db-security-group`

L'exemple de code suivant montre comment utiliser`create-db-security-group`.

AWS CLI

Pour créer un groupe de sécurité de base de données Amazon RDS

La `create-db-security-group` commande suivante crée un nouveau groupe de sécurité de base de données Amazon RDS :

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-group-description "My Test Security Group"
```

Dans l'exemple, le nouveau groupe de sécurité de base de données est nommé `mysecgroup` et possède une description.

Sortie :

```
{
    "DBSecurityGroup": {
        "OwnerId": "123456789012",
        "DBSecurityGroupName": "mysecgroup",
        "DBSecurityGroupDescription": "My Test Security Group",
        "VpcId": "vpc-a1b2c3d4",
        "EC2SecurityGroups": [],
        "IPRanges": [],
        "DBSecurityGroupArn": "arn:aws:rds:us-west-2:123456789012:secgrp:mysecgroup"
    }
}
```

- Pour plus de détails sur l'API, voir CreateDbSecurityGroupla section Référence des AWS CLI commandes.

`create-db-shard-group`

L'exemple de code suivant montre comment utiliser`create-db-shard-group`.

AWS CLI

Exemple 1 : pour créer un cluster de base de données principal Aurora PostgreSQL

L'`create-db-cluster`exemple suivant crée un cluster de base de données principal SQL Aurora PostgreSQL compatible avec Aurora Serverless v2 et Aurora Limitless Database.

```
aws rds create-db-cluster \
    --db-cluster-identifier my-sv2-cluster \
    --engine aurora-postgresql \
    --engine-version 15.2-limitless \
    --storage-type aurora-iopt1 \
    --serverless-v2-scaling-configuration MinCapacity=2,MaxCapacity=16 \
    --enable-limitless-database \
    --master-username myuser \
    --master-user-password mypassword \
    --enable-cloudwatch-logs-exports postgresql
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-2b",
            "us-east-2c",
            "us-east-2a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "my-sv2-cluster",
        "DBClusterParameterGroup": "default.aurora-postgresql15",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "my-sv2-cluster.cluster-cekycexample.us-
east-2.rds.amazonaws.com",
        "ReaderEndpoint": "my-sv2-cluster.cluster-ro-cekycexample.us-
east-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "15.2-limitless",
        "Port": 5432,
        "MasterUsername": "myuser",
        "PreferredBackupWindow": "06:05-06:35",
        "PreferredMaintenanceWindow": "mon:08:25-mon:08:55",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
```

```
            }
        ],
        "HostedZoneId": "Z2XHWR1EXAMPLE",
        "StorageEncrypted": false,
        "DbClusterResourceId": "cluster-XYEDT6ML6FHIXH4Q2J1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-east-2:123456789012:cluster:my-sv2-cluster",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2024-02-19T16:24:07.771000+00:00",
        "EnabledCloudwatchLogsExports": [
            "postgresql"
        ],
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": [],
        "TagList": [],
        "StorageType": "aurora-iopt1",
        "AutoMinorVersionUpgrade": true,
        "ServerlessV2ScalingConfiguration": {
            "MinCapacity": 2.0,
            "MaxCapacity": 16.0
        },
        "NetworkType": "IPV4",
        "IOOptimizedNextAllowedModificationTime":
 "2024-03-21T16:24:07.781000+00:00",
        "LimitlessDatabase": {
            "Status": "not-in-use",
            "MinRequiredACU": 96.0
        }
    }
}
```

Exemple 2 : pour créer l'instance de base de données principale (rédacteur)

L'`create-db-instance`exemple suivant crée une instance de base de données principale (scripteur) Aurora Serverless v2. Lorsque vous utilisez la console pour créer un cluster de base de données, Amazon RDS crée automatiquement l'instance de base de données d'écriture pour votre cluster de base de données. Toutefois, lorsque vous utilisez l' AWS interface de ligne de commande pour créer un cluster de base de données, vous devez créer explicitement l'instance

de base de données d'écriture pour votre cluster de base de données à l'aide de la commande `create-db-instance` AWS CLI.

```
aws rds create-db-instance \
    --db-instance-identifier my-sv2-instance \
    --db-cluster-identifier my-sv2-cluster \
    --engine aurora-postgresql \
    --db-instance-class db.serverless
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-sv2-instance",
        "DBInstanceClass": "db.serverless",
        "Engine": "aurora-postgresql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "myuser",
        "AllocatedStorage": 1,
        "PreferredBackupWindow": "06:05-06:35",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.aurora-postgresql15",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-########",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
```

```
                    "Name": "us-east-2c"
                },
                "SubnetOutpost": {},
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-########",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-2a"
                },
                "SubnetOutpost": {},
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-########",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-2b"
                },
                "SubnetOutpost": {},
                "SubnetStatus": "Active"
            }
        ]
    },
    "PreferredMaintenanceWindow": "fri:09:01-fri:09:31",
    "PendingModifiedValues": {
        "PendingCloudwatchLogsExports": {
            "LogTypesToEnable": [
                "postgresql"
            ]
        }
    },
    "MultiAZ": false,
    "EngineVersion": "15.2-limitless",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "postgresql-license",
    "OptionGroupMemberships": [
        {
            "OptionGroupName": "default:aurora-postgresql-15",
            "Status": "in-sync"
        }
    ],
    "PubliclyAccessible": false,
    "StorageType": "aurora-iopt1",
```

```
        "DbInstancePort": 0,
        "DBClusterIdentifier": "my-sv2-cluster",
        "StorageEncrypted": false,
        "DbiResourceId": "db-BIQTE3B3K3RM7M74SK5EXAMPLE",
        "CACertificateIdentifier": "rds-ca-rsa2048-g1",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-2:123456789012:db:my-sv2-instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": [],
        "TagList": [],
        "CustomerOwnedIpEnabled": false,
        "BackupTarget": "region",
        "NetworkType": "IPV4",
        "StorageThroughput": 0,
        "CertificateDetails": {
            "CAIdentifier": "rds-ca-rsa2048-g1"
        },
        "DedicatedLogVolume": false
    }
}
```

Exemple 3 : pour créer le groupe de partitions de base de données

L'`create-db-shard-group`exemple suivant crée un groupe de partitions de base de données dans votre cluster de base de données principal Aurora PostgreSQL.

```
aws rds create-db-shard-group \
    --db-shard-group-identifier my-db-shard-group \
    --db-cluster-identifier my-sv2-cluster \
    --max-acu 768
```

Sortie :

```
{
    "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
    "DBShardGroupIdentifier": "my-db-shard-group",
    "DBClusterIdentifier": "my-sv2-cluster",
    "MaxACU": 768.0,
```

```
    "ComputeRedundancy": 0,
    "Status": "creating",
    "PubliclyAccessible": false,
    "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-east-2.rds.amazonaws.com"
}
```

Pour plus d'informations, consultez la section Utilisation d'Aurora Serverless v2 dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateDbShardGroupla section Référence des AWS CLI commandes.

`create-db-snapshot`

L'exemple de code suivant montre comment utiliser`create-db-snapshot`.

AWS CLI

Pour créer un instantané de base de données

L'`create-db-snapshot`exemple suivant crée un instantané de base de données.

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

Sortie :

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "creating",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
```

```
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 0,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

Pour plus d'informations, consultez la section Création d'un instantané de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Create DBSnapshot in AWS CLI Command Reference.

`create-db-subnet-group`

L'exemple de code suivant montre comment utiliser`create-db-subnet-group`.

AWS CLI

Pour créer un groupe de sous-réseaux de base de données

L'`create-db-subnet-group`exemple suivant crée un groupe de sous-réseaux de base de données appelé à `mysubnetgroup` l'aide de sous-réseaux existants.

```
aws rds create-db-subnet-group \
    --db-subnet-group-name mysubnetgroup \
    --db-subnet-group-description "test DB subnet group" \
    --subnet-ids
 '["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef"]'
```

Sortie :

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mysubnetgroup",
        "DBSubnetGroupDescription": "test DB subnet group",
```

```
        "VpcId": "vpc-0f08e7610a1b2c3d4",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ],
        "DBSubnetGroupArn": "arn:aws:rds:us-
west-2:0123456789012:subgrp:mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez la section Création d'une instance de base de données dans un VPC dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir CreateDbSubnetGroupla section Référence des AWS CLI commandes.

`create-event-subscription`

L'exemple de code suivant montre comment utiliser`create-event-subscription`.

AWS CLI

Pour créer un abonnement à un événement

L'`create-event-subscription`exemple suivant crée un abonnement pour les événements de sauvegarde et de restauration pour les instances de base de données du AWS compte courant. Les notifications sont envoyées à une rubrique Amazon Simple Notification Service, spécifiée par`--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Sortie :

```
{
    "EventSubscription": {
        "Status": "creating",
        "CustSubscriptionId": "my-instance-events",
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "CustomerAwsId": "123456789012",
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "SourceType": "db-instance",
        "Enabled": true
    }
}
```

- Pour plus de détails sur l'API, voir CreateEventSubscriptionla section Référence des AWS CLI commandes.

`create-global-cluster`

L'exemple de code suivant montre comment utiliser`create-global-cluster`.

AWS CLI

Pour créer un cluster de bases de données global

L'`create-global-cluster`exemple suivant crée un nouveau cluster de base de données global compatible Aurora MySQL.

```
aws rds create-global-cluster \
    --global-cluster-identifier myglobalcluster \
    --engine aurora-mysql
```

Sortie :

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
        "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.mysql_aurora.2.07.2",
        "StorageEncrypted": false,
        "DeletionProtection": false,
        "GlobalClusterMembers": []
    }
}
```

Pour plus d'informations, consultez la section Création d'une base de données globale Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreateGlobalClusterla section Référence des AWS CLI commandes.

`create-option-group`

L'exemple de code suivant montre comment utiliser`create-option-group`.

AWS CLI

Pour créer un groupe d'options Amazon RDS

La `create-option-group` commande suivante crée un nouveau groupe d'options Amazon RDS pour la `Oracle Enterprise Edition` version `11.2`, is named ``MyOptionGroup` et inclut une description.

```
aws rds create-option-group \
    --option-group-name MyOptionGroup \
    --engine-name oracle-ee \
    --major-engine-version 11.2 \
    --option-group-description "Oracle Database Manager Database Control"
```

Sortie :

```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Oracle Database Manager Database Control",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "11.2",
        "Options": [],
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "OptionGroupArn": "arn:aws:rds:us-west-2:123456789012:og:myoptiongroup"
    }
}
```

- Pour plus de détails sur l'API, voir CreateOptionGroupla section Référence des AWS CLI commandes.

`delete-blue-green-deployment`

L'exemple de code suivant montre comment utiliser`delete-blue-green-deployment`.

AWS CLI

Exemple 1 : Pour supprimer des ressources dans un environnement vert pour une instance de base de données RDS pour MySQL

L'`delete-blue-green-deployment`exemple suivant supprime les ressources dans un environnement écologique pour une instance de base de données RDS pour MySQL.

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-v53303651eexfake \
    --delete-target
```

Sortie :

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
        "BlueGreenDeploymentName": "bgd-cli-test-instance",
        "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
rkfbpe",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-rkfbpe",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-j382ha",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ejv4ao",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-vlpz3t",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
```

```
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CONFIGURE_BACKUPS",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                "Status": "COMPLETED"
            }
        ],
        "Status": "DELETING",
        "CreateTime": "2022-02-25T21:18:51.183000+00:00",
        "DeleteTime": "2022-02-25T22:25:31.331000+00:00"
    }
}
```

Pour plus d'informations, consultez Supprimer un blue/green déploiement dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour supprimer des ressources dans un environnement vert pour un cluster de base de données Aurora MySQL

L'`delete-blue-green-deployment`exemple suivant supprime les ressources dans un environnement écologique pour un cluster de base de données Aurora MySQL.

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --delete-target
```

Sortie :

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
        "SwitchoverDetails": [
            {
```

```
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-gpmaxf",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-j2oajq",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-mkxies",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
```

```
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_CUSTOM_ENDPOINTS",
                "Status": "COMPLETED"
            }
        ],
        "Status": "DELETING",
        "CreateTime": "2022-02-25T21:12:00.288000+00:00",
        "DeleteTime": "2022-02-25T22:29:11.336000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un blue/green déploiement dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteBlueGreenDeploymentla section Référence des AWS CLI commandes.

`delete-db-cluster-endpoint`

L'exemple de code suivant montre comment utiliser`delete-db-cluster-endpoint`.

AWS CLI

Pour supprimer un point de terminaison de cluster de base de données personnalisé

L'`delete-db-cluster-endpoint`exemple suivant supprime le point de terminaison du cluster de base de données personnalisé spécifié.

```
aws rds delete-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint
```

Sortie :

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "deleting",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2",
        "dbinstance3"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

Pour plus d'informations, consultez la section Gestion des connexions Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbClusterEndpointla section Référence des AWS CLI commandes.

`delete-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-db-cluster-parameter-group`.

AWS CLI

Pour supprimer un groupe de paramètres de cluster de base de données

L'`delete-db-cluster-parameter-group`exemple suivant supprime le groupe de paramètres de cluster de base de données spécifié.

```
aws rds delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbClusterParameterGroupla section Référence des AWS CLI commandes.

`delete-db-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`delete-db-cluster-snapshot`.

AWS CLI

Pour supprimer un instantané de cluster de base de données

L'`delete-db-cluster-snapshot`exemple suivant supprime le snapshot du cluster de base de données spécifié.

```
aws rds delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier mydbclustersnapshot
```

Sortie :

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "DBClusterSnapshotIdentifier": "mydbclustersnapshot",
        "DBClusterIdentifier": "mydbcluster",
        "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 0,
        "Status": "available",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
```

```
        "SnapshotType": "manual",
        "PercentProgress": 100,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:mydbclustersnapshot",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

Pour plus d'informations, consultez Supprimer un instantané dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbClusterSnapshotla section Référence des AWS CLI commandes.

`delete-db-cluster`

L'exemple de code suivant montre comment utiliser`delete-db-cluster`.

AWS CLI

Exemple 1 : pour supprimer une instance de base de données dans un cluster de base de données

L'`delete-db-instance`exemple suivant supprime l'instance de base de données finale dans un cluster de base de données. Vous ne pouvez pas supprimer un cluster de base de données s'il contient des instances de base de données qui ne sont pas dans l'état de suppression. Vous ne pouvez pas prendre un instantané final lors de la suppression d'une instance de base de données dans un cluster de base de données.

```
aws rds delete-db-instance \
    --db-instance-identifier database-3
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-3",
        "DBInstanceClass": "db.r4.large",
        "Engine": "aurora-postgresql",
```

```
        "DBInstanceStatus": "deleting",

    ...output omitted...

    }
}
```

Pour plus d'informations, consultez Supprimer une instance de base de données dans un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : pour supprimer un cluster de base de données

L'`delete-db-cluster`exemple suivant supprime le cluster de base de données nommé `mycluster` et prend un instantané final nommé`mycluster-final-snapshot`. L'état du cluster de base de données est disponible pendant la prise du snapshot. Pour suivre la progression de la suppression, utilisez la commande `describe-db-clusters` CLI.

```
aws rds delete-db-cluster \
    --db-cluster-identifier mycluster \
    --no-skip-final-snapshot \
    --final-db-snapshot-identifier mycluster-final-snapshot
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 20,
        "AvailabilityZones": [
            "eu-central-1b",
            "eu-central-1c",
            "eu-central-1a"
        ],
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "mycluster",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default-vpc-aa11bb22",
        "Status": "available",

    ...output omitted...

    }
}
```

Pour plus d'informations, consultez Clusters Aurora avec une seule instance de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbClusterla section Référence des AWS CLI commandes.

`delete-db-instance-automated-backup`

L'exemple de code suivant montre comment utiliser`delete-db-instance-automated-backup`.

AWS CLI

Pour supprimer une sauvegarde automatique répliquée d'une région

L'`delete-db-instance-automated-backup`exemple suivant supprime la sauvegarde automatique avec le nom de ressource Amazon (ARN) spécifié.

```
aws rds delete-db-instance-automated-backup \
    --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
```

Sortie :

```
{
    "DBInstanceAutomatedBackup": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
        "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
        "Region": "us-east-1",
        "DBInstanceIdentifier": "new-orcl-db",
        "RestoreWindow": {},
        "AllocatedStorage": 20,
        "Status": "deleting",
        "Port": 1521,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-########",
        "InstanceCreateTime": "2020-12-04T15:28:31Z",
        "MasterUsername": "admin",
        "Engine": "oracle-se2",
        "EngineVersion": "12.1.0.2.v21",
        "LicenseModel": "bring-your-own-license",
        "OptionGroupName": "default:oracle-se2-12-1",
        "Encrypted": false,
```

```
        "StorageType": "gp2",
        "IAMDatabaseAuthenticationEnabled": false,
        "BackupRetentionPeriod": 7,
        "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

Pour plus d'informations, consultez Supprimer des sauvegardes répliquées dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir DeleteDbInstanceAutomatedBackupla section Référence des AWS CLI commandes.

`delete-db-instance`

L'exemple de code suivant montre comment utiliser`delete-db-instance`.

AWS CLI

Pour supprimer une instance de base de données

L'`delete-db-instance`exemple suivant supprime l'instance de base de données spécifiée après avoir créé un instantané de base de données final nommé`test-instance-final-snap`.

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- Pour plus de détails sur l'API, voir Supprimer DBInstance dans le manuel de référence des AWS CLI commandes.

`delete-db-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-db-parameter-group`.

AWS CLI

Pour supprimer un groupe de paramètres de base de données

L'`command`exemple suivant supprime un groupe de paramètres de base de données.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

Cette commande ne produit aucun résultat.

Pour de plus amples informations, veuillez consulter Utilisation des groupes de paramètres de base de données dans le Guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Supprimer le DBParameter groupe dans AWS CLI la référence des commandes.

`delete-db-proxy-endpoint`

L'exemple de code suivant montre comment utiliser`delete-db-proxy-endpoint`.

AWS CLI

Pour supprimer un point de terminaison proxy de base de données pour une base de données RDS

L'`delete-db-proxy-endpoint`exemple suivant supprime un point de terminaison de proxy de base de données pour la base de données cible.

```
aws rds delete-db-proxy-endpoint \
    --db-proxy-endpoint-name proxyEP1
```

Sortie :

```
{
"DBProxyEndpoint":
    {
        "DBProxyEndpointName": "proxyEP1",
```

```
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
        "DBProxyName": "proxyExample",
        "Status": "deleting",
        "VpcId": "vpc-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234",
            "sg-5678"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyEP1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "CreatedDate": "2023-04-13T01:49:38.568000+00:00",
        "TargetRole": "READ_ONLY",
        "IsDefault": false
    }
}
```

Pour plus d'informations, consultez Supprimer un point de terminaison de proxy dans le guide de l'utilisateur Amazon RDS et Suppression d'un point de terminaison de proxy dans le guide de l'utilisateur d'Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbProxyEndpointla section Référence des AWS CLI commandes.

`delete-db-proxy`

L'exemple de code suivant montre comment utiliser`delete-db-proxy`.

AWS CLI

Pour supprimer un proxy de base de données pour une base de données RDS

L'`delete-db-proxy`exemple suivant supprime un proxy de base de données.

```
aws rds delete-db-proxy \
    --db-proxy-name proxyExample
```

Sortie :

```
{
        "DBProxy":
        {
            "DBProxyName": "proxyExample",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
            "Status": "deleting",
            "EngineFamily": "PostgreSQL",
            "VpcId": "vpc-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234",
                "sg-5678"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                "Description": "proxydescription`"
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
                } ],
            "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

Pour plus d'informations, consultez Supprimer un proxy RDS dans le guide de l'utilisateur Amazon RDS et Suppression d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbProxyla section Référence des AWS CLI commandes.

`delete-db-security-group`

L'exemple de code suivant montre comment utiliser`delete-db-security-group`.

AWS CLI

Pour supprimer un groupe de sécurité de base de données

L'`delete-db-security-group`exemple suivant supprime un groupe de sécurité de base de données nommé`mysecuritygroup`.

```
aws rds delete-db-security-group \
    --db-security-group-name mysecuritygroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec des groupes de sécurité de base de données (plateforme EC2 classique) dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir DeleteDbSecurityGroupla section Référence des AWS CLI commandes.

`delete-db-shard-group`

L'exemple de code suivant montre comment utiliser`delete-db-shard-group`.

AWS CLI

Exemple 1 : pour supprimer un groupe de partitions de base de données sans succès

L'`delete-db-shard-group`exemple suivant montre l'erreur qui se produit lorsque vous essayez de supprimer un groupe de partitions de base de données avant de supprimer l'ensemble de vos bases de données et de vos schémas.

```
aws rds delete-db-shard-group \
    --db-shard-group-identifier limitless-test-shard-grp
```

Sortie :

```
An error occurred (InvalidDBShardGroupState) when calling the DeleteDBShardGroup
 operation: Unable to delete the DB shard group limitless-test-db-shard-group.
```

```
Delete all of your Limitless Database databases and schemas, then try again.
```

Exemple 2 : pour supprimer un groupe de partitions de base de données avec succès

L'`delete-db-shard-group`exemple suivant supprime un groupe de partitions de base de données une fois que vous avez supprimé toutes vos bases de données et tous vos schémas, y compris le schéma. `public`

```
aws rds delete-db-shard-group \
    --db-shard-group-identifier limitless-test-shard-grp
```

Sortie :

```
{
    "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
    "DBShardGroupIdentifier": "limitless-test-shard-grp",
    "DBClusterIdentifier": "limitless-test-cluster",
    "MaxACU": 768.0,
    "ComputeRedundancy": 0,
    "Status": "deleting",
    "PubliclyAccessible": true,
    "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
}
```

Pour plus d'informations, consultez la section Suppression de clusters de base de données et d'instances de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteDbShardGroupla section Référence des AWS CLI commandes.

`delete-db-snapshot`

L'exemple de code suivant montre comment utiliser`delete-db-snapshot`.

AWS CLI

Pour supprimer un instantané de base de données

L'`delete-db-snapshot`exemple suivant supprime le snapshot de base de données spécifié.

```
aws rds delete-db-snapshot \
```

```
    --db-snapshot-identifier mydbsnapshot
```

Sortie :

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "SnapshotCreateTime": "2019-06-18T22:08:40.702Z",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "deleted",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 100,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

Pour plus d'informations, consultez Supprimer un instantané dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir DeleteDbSnapshotla section Référence des AWS CLI commandes.

`delete-db-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-db-subnet-group`.

AWS CLI

Pour supprimer un groupe de sous-réseaux de base de données

L'`delete-db-subnet-group`exemple suivant supprime le groupe de sous-réseaux de base de données appelé. `mysubnetgroup`

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec une instance de base de données dans un VPC dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir DeleteDbSubnetGroupla section Référence des AWS CLI commandes.

`delete-event-subscription`

L'exemple de code suivant montre comment utiliser`delete-event-subscription`.

AWS CLI

Pour supprimer un abonnement à un événement

L'`delete-event-subscription`exemple suivant supprime l'abonnement à l'événement spécifié.

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

Sortie :

```
{
    "EventSubscription": {
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "CustomerAwsId": "123456789012",
        "Enabled": false,
        "SourceIdsList": [
            "test-instance"
        ],
        "SourceType": "db-instance",
```

```
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
        "CustSubscriptionId": "my-instance-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "Status": "deleting"
    }
}
```

- Pour plus de détails sur l'API, voir DeleteEventSubscriptionla section Référence des AWS CLI commandes.

`delete-global-cluster`

L'exemple de code suivant montre comment utiliser`delete-global-cluster`.

AWS CLI

Pour supprimer un cluster de bases de données global

L'`delete-global-cluster`exemple suivant supprime un cluster de base de données global compatible Aurora MySQL. La sortie indique le cluster que vous supprimez, mais `describe-global-clusters` les commandes suivantes ne répertorient pas ce cluster de base de données.

```
aws rds delete-global-cluster \
    --global-cluster-identifier myglobalcluster
```

Sortie :

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
        "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.mysql_aurora.2.07.2",
        "StorageEncrypted": false,
```

```
        "DeletionProtection": false,
        "GlobalClusterMembers": []
    }
}
```

Pour plus d'informations, consultez Supprimer une base de données globale Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeleteGlobalClusterla section Référence des AWS CLI commandes.

`delete-option-group`

L'exemple de code suivant montre comment utiliser`delete-option-group`.

AWS CLI

Pour supprimer un groupe d'options

L'`delete-option-group`exemple suivant supprime le groupe d'options spécifié.

```
aws rds delete-option-group \
    --option-group-name myoptiongroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un groupe d'options dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir DeleteOptionGroupla section Référence des AWS CLI commandes.

`deregister-db-proxy-targets`

L'exemple de code suivant montre comment utiliser`deregister-db-proxy-targets`.

AWS CLI

Pour désenregistrer une cible proxy de base de données du groupe cible de base de données

L'`deregister-db-proxy-targets`exemple suivant supprime l'association entre le proxy `proxyExample` et sa cible.

```
aws rds deregister-db-proxy-targets \
    --db-proxy-name proxyExample \
    --db-instance-identifiers database-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Supprimer un proxy RDS dans le guide de l'utilisateur Amazon RDS et Suppression d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeregisterDbProxyTargetsla section Référence des AWS CLI commandes.

`describe-account-attributes`

L'exemple de code suivant montre comment utiliser`describe-account-attributes`.

AWS CLI

Pour décrire les attributs du compte

L'`describe-account-attributes`exemple suivant récupère les attributs du AWS compte courant.

```
aws rds describe-account-attributes
```

Sortie :

```
{
    "AccountQuotas": [
        {
            "Max": 40,
            "Used": 4,
            "AccountQuotaName": "DBInstances"
        },
        {
            "Max": 40,
            "Used": 0,
            "AccountQuotaName": "ReservedDBInstances"
        },
        {
            "Max": 100000,
            "Used": 40,
```

```
            "AccountQuotaName": "AllocatedStorage"
        },
        {
            "Max": 25,
            "Used": 0,
            "AccountQuotaName": "DBSecurityGroups"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBParameterGroups"
        },
        {
            "Max": 100,
            "Used": 3,
            "AccountQuotaName": "ManualSnapshots"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "EventSubscriptions"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBSubnetGroups"
        },
        {
            "Max": 20,
            "Used": 1,
            "AccountQuotaName": "OptionGroups"
        },
        {
            "Max": 20,
            "Used": 6,
            "AccountQuotaName": "SubnetsPerDBSubnetGroup"
        },
        {
            "Max": 5,
```

```
            "Used": 0,
            "AccountQuotaName": "ReadReplicasPerMaster"
        },
        {
            "Max": 40,
            "Used": 1,
            "AccountQuotaName": "DBClusters"
        },
        {
            "Max": 50,
            "Used": 0,
            "AccountQuotaName": "DBClusterParameterGroups"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "DBClusterRoles"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeAccountAttributesla section Référence des AWS CLI commandes.

`describe-blue-green-deployments`

L'exemple de code suivant montre comment utiliser`describe-blue-green-deployments`.

AWS CLI

Exemple 1 : pour décrire le blue/green déploiement d'une instance de base de données RDS une fois la création terminée

L'`describe-blue-green-deployment`exemple suivant permet de récupérer les détails d'un blue/green déploiement une fois sa création terminée.

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-v53303651eexfake
```

Sortie :

```
{
```

```
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
            "BlueGreenDeploymentName": "bgd-cli-test-instance",
            "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
            "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
rkfbpe",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-rkfbpe",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-j382ha",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ejv4ao",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-vlpz3t",
                    "Status": "AVAILABLE"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
```

```
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CONFIGURE_BACKUPS",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "AVAILABLE",
            "CreateTime": "2022-02-25T21:18:51.183000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation d'un blue/green déploiement dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : Pour décrire un blue/green déploiement pour un cluster de base de données Aurora MySQL

L'`describe-blue-green-deployment`exemple suivant permet de récupérer les détails d'un blue/green déploiement.

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

Sortie :

```
{
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
            "BlueGreenDeploymentName": "my-blue-green-deployment",
            "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
            "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
            "SwitchoverDetails": [
                {
```

```
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-green-gpmaxf",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-green-j2oajq",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-green-mkxies",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                    "Status": "AVAILABLE"
                }
            ],
            "Tasks": [
                {
```

```
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_CUSTOM_ENDPOINTS",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "AVAILABLE",
            "CreateTime": "2022-02-25T21:12:00.288000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation d'un blue/green déploiement dans le guide de l'utilisateur Amazon Aurora.

Exemple 3 : pour décrire un blue/green déploiement pour un cluster Aurora MySQL après le basculement

L'`describe-blue-green-deployment`exemple suivant permet de récupérer les détails d'un blue/green déploiement une fois que l'environnement écologique a été promu environnement de production.

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

Sortie :

```
{
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
            "BlueGreenDeploymentName": "my-blue-green-deployment",
```

```
            "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-old1",
            "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-old1",
```

```
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                    "Status": "SWITCHOVER_COMPLETED"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_CUSTOM_ENDPOINTS",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "SWITCHOVER_COMPLETED",
            "CreateTime": "2022-02-25T22:38:49.522000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Visualisation d'un blue/green déploiement dans le guide de l'utilisateur Amazon Aurora.

Exemple 4 : pour décrire un blue/green déploiement combiné

L'`describe-blue-green-deployment`exemple suivant permet de récupérer les détails d'un blue/green déploiement combiné.

```
aws rds describe-blue-green-deployments
```

Sortie :

```
{
    "BlueGreenDeployments": [
```

```
        {
            "BlueGreenDeploymentIdentifier": "bgd-wi89nwzgfakelccs",
            "BlueGreenDeploymentName": "my-blue-green-deployment",
            "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
            "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-green-gpmaxf",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-green-j2oajq",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-green-mkxies",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
                    "Status": "AVAILABLE"
                },
                {
```

```
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                    "Status": "AVAILABLE"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_CUSTOM_ENDPOINTS",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "AVAILABLE",
            "CreateTime": "2022-02-25T21:12:00.288000+00:00"
        },
        {
            "BlueGreenDeploymentIdentifier": "bgd-v5330365fake1eex",
            "BlueGreenDeploymentName": "bgd-cli-test-instance",
            "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-old1",
            "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-old1",
```

```
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                    "Status": "SWITCHOVER_COMPLETED"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CONFIGURE_BACKUPS",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "SWITCHOVER_COMPLETED",
            "CreateTime": "2022-02-25T22:33:22.225000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez les sections Affichage d'un blue/green déploiement dans le guide de l'utilisateur Amazon RDS et Affichage d'un blue/green déploiement dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeBlueGreenDeploymentsà la section Référence des AWS CLI commandes.

`describe-certificates`

L'exemple de code suivant montre comment utiliser`describe-certificates`.

AWS CLI

Pour décrire les certificats

L'`describe-certificates`exemple suivant récupère les détails du certificat associé à la région par défaut de l'utilisateur.

```
aws rds describe-certificates
```

Sortie :

```
{
    "Certificates": [
        {
            "CertificateIdentifier": "rds-ca-ecc384-g1",
            "CertificateType": "CA",
            "Thumbprint": "2ee3dcc06e50192559b13929e73484354f23387d",
            "ValidFrom": "2021-05-24T22:06:59+00:00",
            "ValidTill": "2121-05-24T23:06:59+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-ecc384-g1",
            "CustomerOverride": false
        },
        {
            "CertificateIdentifier": "rds-ca-rsa4096-g1",
            "CertificateType": "CA",
            "Thumbprint": "19da4f2af579a8ae1f6a0fa77aa5befd874b4cab",
            "ValidFrom": "2021-05-24T22:03:20+00:00",
            "ValidTill": "2121-05-24T23:03:20+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa4096-g1",
            "CustomerOverride": false
        },
```

```
        {
            "CertificateIdentifier": "rds-ca-rsa2048-g1",
            "CertificateType": "CA",
            "Thumbprint": "7c40cb42714b6fdb2b296f9bbd0e8bb364436a76",
            "ValidFrom": "2021-05-24T21:59:00+00:00",
            "ValidTill": "2061-05-24T22:59:00+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa2048-g1",
            "CustomerOverride": true,
            "CustomerOverrideValidTill": "2061-05-24T22:59:00+00:00"
        },
        {
            "CertificateIdentifier": "rds-ca-2019",
            "CertificateType": "CA",
            "Thumbprint": "d40ddb29e3750dffa671c3140bbf5f478d1c8096",
            "ValidFrom": "2019-08-22T17:08:50+00:00",
            "ValidTill": "2024-08-22T17:08:50+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-2019",
            "CustomerOverride": false
        }
    ],
    "DefaultCertificateForNewLaunches": "rds-ca-rsa2048-g1"
}
```

Pour plus d'informations, consultez les sections Utilisation SSL/TLS pour chiffrer une connexion à une instance de base de données dans le guide de l'utilisateur Amazon RDS et Utilisation SSL/TLS pour chiffrer une connexion à un cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeCertificatesà la section Référence des AWS CLI commandes.

`describe-db-cluster-backtracks`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-backtracks`.

AWS CLI

Pour décrire les retours en arrière d'un cluster de bases de données

L'`describe-db-cluster-backtracks`exemple suivant récupère les détails du cluster de base de données spécifié.

```
aws rds describe-db-cluster-backtracks \
```

```
    --db-cluster-identifier mydbcluster
```

Sortie :

```
{
    "DBClusterBacktracks": [
        {
            "DBClusterIdentifier": "mydbcluster",
            "BacktrackIdentifier": "2f5f5294-0dd2-44c9-9f50-EXAMPLE",
            "BacktrackTo": "2021-02-12T04:59:22Z",
            "BacktrackedFrom": "2021-02-12T14:37:31.640Z",
            "BacktrackRequestCreationTime": "2021-02-12T14:36:18.819Z",
            "Status": "COMPLETED"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "BacktrackIdentifier": "3c7a6421-af2a-4ea3-ae95-EXAMPLE",
            "BacktrackTo": "2021-02-11T22:53:46Z",
            "BacktrackedFrom": "2021-02-12T00:09:27.006Z",
            "BacktrackRequestCreationTime": "2021-02-12T00:07:53.487Z",
            "Status": "COMPLETED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Suivi d'un cluster de base de données Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterBacktracksà la section Référence des AWS CLI commandes.

`describe-db-cluster-endpoints`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-endpoints`.

AWS CLI

Exemple 1 : pour décrire les points de terminaison d'un cluster de bases de données

L'`describe-db-cluster-endpoints`exemple suivant permet de récupérer les détails des points de terminaison de votre cluster de base de données. Les types de clusters Aurora les plus courants possèdent deux points de terminaison. Un point de terminaison possède un

typeWRITER. Vous pouvez utiliser ce point de terminaison pour toutes les instructions SQL. L'autre point de terminaison est de typeREADER. Vous ne pouvez utiliser ce point de terminaison que pour SELECT et les autres instructions SQL en lecture seule.

```
aws rds describe-db-cluster-endpoints
```

Sortie :

```
{
    "DBClusterEndpoints": [
        {
            "DBClusterIdentifier": "my-database-1",
            "Endpoint": "my-database-1.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "creating",
            "EndpointType": "WRITER"
        },
        {
            "DBClusterIdentifier": "my-database-1",
            "Endpoint": "my-database-1.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "creating",
            "EndpointType": "READER"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "Endpoint": "mydbcluster.cluster-cnpexamle.us-east-1.rds.amazonaws.com",
            "Status": "available",
            "EndpointType": "WRITER"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "Endpoint": "mydbcluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "available",
            "EndpointType": "READER"
        }
    ]
}
```

Exemple 2 : pour décrire les points de terminaison d'un cluster de base de données unique

L'`describe-db-cluster-endpoints`exemple suivant récupère les détails des points de terminaison du cluster de base de données d'un seul cluster de base de données spécifié. Les clusters Aurora Serverless ne possèdent qu'un seul point de terminaison de `WRITER` type.

```
aws rds describe-db-cluster-endpoints \
    --db-cluster-identifier serverless-cluster
```

Sortie :

```
{
    "DBClusterEndpoints": [
        {
            "Status": "available",
            "Endpoint": "serverless-cluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "DBClusterIdentifier": "serverless-cluster",
            "EndpointType": "WRITER"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des connexions Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterEndpointsà la section Référence des AWS CLI commandes.

`describe-db-cluster-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-parameter-groups`.

AWS CLI

Pour décrire les groupes de paramètres du cluster de base de données

L'`describe-db-cluster-parameter-groups`exemple suivant récupère les détails de vos groupes de paramètres de cluster de base de données.

```
aws rds describe-db-cluster-parameter-groups
```

Sortie :

```
{
    "DBClusterParameterGroups": [
        {
            "DBClusterParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default cluster parameter group for aurora-mysql5.7",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora-mysql5.7"
        },
        {
            "DBClusterParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default cluster parameter group for aurora-
postgresql9.6",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora-postgresql9.6"
        },
        {
            "DBClusterParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default cluster parameter group for aurora5.6",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora5.6"
        },
        {
            "DBClusterParameterGroupName": "mydbclusterpg",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "My DB cluster parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpg"
        },
        {
            "DBClusterParameterGroupName": "mydbclusterpgcopy",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Copy of mydbclusterpg parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpgcopy"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterParameterGroupsà la section Référence des AWS CLI commandes.

`describe-db-cluster-parameters`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-parameters`.

AWS CLI

Exemple 1 : pour décrire les paramètres d'un groupe de paramètres de cluster de base de données

L'`describe-db-cluster-parameters`exemple suivant récupère des détails sur les paramètres d'un groupe de paramètres de cluster de base de données.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name mydbclusterpg
```

Sortie :

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        {
```

```
            "ParameterName": "aurora_lab_mode",
            "ParameterValue": "0",
            "Description": "Enables new features in the Aurora engine.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        ...some output truncated...
    ]
}
```

Exemple 2 : pour répertorier uniquement les noms de paramètres d'un groupe de paramètres de cluster de base de données

L'`describe-db-cluster-parameters`exemple suivant récupère uniquement les noms des paramètres d'un groupe de paramètres de cluster de base de données.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].{ParameterName:ParameterName}'
```

Sortie :

```
[
    {
        "ParameterName": "allow-suspicious-udfs"
    },
    {
        "ParameterName": "aurora_binlog_read_buffer_size"
    },
    {
        "ParameterName": "aurora_binlog_replication_max_yield_seconds"
    },
    {
        "ParameterName": "aurora_binlog_use_large_read_buffer"
    },
```

```
    {
        "ParameterName": "aurora_lab_mode"
    },

    ...some output truncated...
    }
]
```

Exemple 3 : pour décrire uniquement les paramètres modifiables d'un groupe de paramètres de cluster de base de données

L'`describe-db-cluster-parameters`exemple suivant récupère uniquement les noms des paramètres que vous pouvez modifier dans un groupe de paramètres de cluster de base de données.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].{ParameterName:ParameterName,IsModifiable:IsModifiable} |
 [?IsModifiable == `true`]'
```

Sortie :

```
[
    {
        "ParameterName": "aurora_binlog_read_buffer_size",
        "IsModifiable": true
    },
    {
        "ParameterName": "aurora_binlog_replication_max_yield_seconds",
        "IsModifiable": true
    },
    {
        "ParameterName": "aurora_binlog_use_large_read_buffer",
        "IsModifiable": true
    },
    {
        "ParameterName": "aurora_lab_mode",
        "IsModifiable": true
    },

    ...some output truncated...
    }
```

```
]
```

Exemple 4 : pour décrire uniquement les paramètres booléens modifiables dans un groupe de paramètres de cluster de base de données

L'`describe-db-cluster-parameters`exemple suivant récupère uniquement les noms des paramètres que vous pouvez modifier dans un groupe de paramètres de cluster de base de données et qui ont un type de données booléen.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].
{ParameterName:ParameterName,DataType:DataType,IsModifiable:IsModifiable} | [?
DataType == `boolean`] | [?IsModifiable == `true`]'
```

Sortie :

```
[
    {
        "DataType": "boolean",
        "ParameterName": "aurora_binlog_use_large_read_buffer",
        "IsModifiable": true
    },
    {
        "DataType": "boolean",
        "ParameterName": "aurora_lab_mode",
        "IsModifiable": true
    },
    {
        "DataType": "boolean",
        "ParameterName": "autocommit",
        "IsModifiable": true
    },
    {
        "DataType": "boolean",
        "ParameterName": "automatic_sp_privileges",
        "IsModifiable": true
    },
    ...some output truncated...
    }
]
```

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterParametersà la section Référence des AWS CLI commandes.

`describe-db-cluster-snapshot-attributes`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-snapshot-attributes`.

AWS CLI

Pour décrire les noms et les valeurs des attributs d'un instantané de cluster de base de données

L'`describe-db-cluster-snapshot-attributes`exemple suivant récupère les détails des noms et des valeurs des attributs pour le cliché du cluster de base de données spécifié.

```
aws rds describe-db-cluster-snapshot-attributes \
    --db-cluster-snapshot-identifier myclustersnapshot
```

Sortie :

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "myclustersnapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012"
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Partage d'un instantané de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterSnapshotAttributesà la section Référence des AWS CLI commandes.

`describe-db-cluster-snapshots`

L'exemple de code suivant montre comment utiliser`describe-db-cluster-snapshots`.

AWS CLI

Pour décrire un instantané de cluster de base de données pour un cluster de base de données

L'`describe-db-cluster-snapshots`exemple suivant récupère les détails des instantanés du cluster de base de données pour le cluster de base de données spécifié.

```
aws rds describe-db-cluster-snapshots \
    --db-cluster-identifier mydbcluster
```

Sortie :

```
{
    "DBClusterSnapshots": [
        {
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1e"
            ],
            "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
            "DBClusterIdentifier": "mydbcluster",
            "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
            "Engine": "aurora-mysql",
            "AllocatedStorage": 0,
            "Status": "available",
            "Port": 0,
            "VpcId": "vpc-6594f31c",
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
            "MasterUsername": "myadmin",
            "EngineVersion": "5.7.mysql_aurora.2.04.2",
            "LicenseModel": "aurora-mysql",
            "SnapshotType": "manual",
            "PercentProgress": 100,
            "StorageEncrypted": true,
```

```
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
            "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:814387698303:cluster-
snapshot:myclustersnapshotcopy",
            "IAMDatabaseAuthenticationEnabled": false
        },
        {
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1e"
            ],
            "DBClusterSnapshotIdentifier": "rds:mydbcluster-2019-06-20-09-16",
            "DBClusterIdentifier": "mydbcluster",
            "SnapshotCreateTime": "2019-06-20T09:16:26.569Z",
            "Engine": "aurora-mysql",
            "AllocatedStorage": 0,
            "Status": "available",
            "Port": 0,
            "VpcId": "vpc-6594f31c",
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
            "MasterUsername": "myadmin",
            "EngineVersion": "5.7.mysql_aurora.2.04.2",
            "LicenseModel": "aurora-mysql",
            "SnapshotType": "automated",
            "PercentProgress": 100,
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
            "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:rds:mydbcluster-2019-06-20-09-16",
            "IAMDatabaseAuthenticationEnabled": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Création d'un instantané de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClusterSnapshotsà la section Référence des AWS CLI commandes.

`describe-db-clusters`

L'exemple de code suivant montre comment utiliser`describe-db-clusters`.

AWS CLI

Exemple 1 : pour décrire un cluster de base de données

L'`describe-db-clusters`exemple suivant récupère les détails du cluster de base de données spécifié.

```
aws rds describe-db-clusters \
    --db-cluster-identifier mydbcluster
```

Sortie :

```
{
    "DBClusters": [
        {
            "AllocatedStorage": 1,
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1e"
            ],
            "BackupRetentionPeriod": 1,
            "DatabaseName": "mydbcluster",
            "DBClusterIdentifier": "mydbcluster",
            "DBClusterParameterGroup": "default.aurora-mysql5.7",
            "DBSubnetGroup": "default",
            "Status": "available",
            "EarliestRestorableTime": "2019-06-19T09:16:28.210Z",
            "Endpoint": "mydbcluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "ReaderEndpoint": "mydbcluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "MultiAZ": true,
            "Engine": "aurora-mysql",
            "EngineVersion": "5.7.mysql_aurora.2.04.2",
            "LatestRestorableTime": "2019-06-20T22:38:14.908Z",
            "Port": 3306,
            "MasterUsername": "myadmin",
            "PreferredBackupWindow": "09:09-09:39",
```

```
            "PreferredMaintenanceWindow": "sat:04:09-sat:04:39",
            "ReadReplicaIdentifiers": [],
            "DBClusterMembers": [
                {
                    "DBInstanceIdentifier": "dbinstance3",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "dbinstance1",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "dbinstance2",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "mydbcluster",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "mydbcluster-us-east-1b",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "mydbcluster",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                }
            ],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
                    "Status": "active"
```

```
                }
            ],
            "HostedZoneId": "Z2R2ITUGPM61AM",
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
            "DbClusterResourceId": "cluster-AKIAIOSFODNN7EXAMPLE",
            "DBClusterArn": "arn:aws:rds:us-
east-1:123456789012:cluster:mydbcluster",
            "AssociatedRoles": [],
            "IAMDatabaseAuthenticationEnabled": false,
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
            "EngineMode": "provisioned",
            "DeletionProtection": false,
            "HttpEndpointEnabled": false
        }
    ]
}
```

Exemple 2 : pour répertorier certains attributs de tous les clusters de base de données

L'`describe-db-clusters`exemple suivant récupère uniquement les `ReaderEndpoint` attributs`DBClusterIdentifier`,`Endpoint`, et de tous vos clusters de base de données dans la AWS région actuelle.

```
aws rds describe-db-clusters \
    --query 'DBClusters[].
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}'
```

Sortie :

```
[
    {
        "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "cluster-57-2020-05-01-2270"
    },
    {
        "Endpoint": "cluster-57-2020-05-01-4615.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
```

```
        "ReaderEndpoint": "cluster-57-2020-05-01-4615.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "cluster-57-2020-05-01-4615"
    },
    {
        "Endpoint": "pg2-cluster.cluster-cnpexample.us-east-1.rds.amazonaws.com",
        "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "pg2-cluster"
    },
    ...output omitted...
    }
]
```

Exemple 3 : pour répertorier les clusters de base de données avec un attribut spécifique

L'`describe-db-clusters`exemple suivant récupère uniquement les `Engine` attributs `DBClusterIdentifier` et de vos clusters de base de données qui utilisent le moteur de `aurora-postgresql` base de données.

```
aws rds describe-db-clusters \
    --query 'DBClusters[].{DBClusterIdentifier:DBClusterIdentifier,Engine:Engine} |
 [?Engine == `aurora-postgresql`]'
```

Sortie :

```
[
    {
        "Engine": "aurora-postgresql",
        "DBClusterIdentifier": "pg2-cluster"
    }
]
```

Pour plus d'informations, consultez Amazon Aurora DB Clusters dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbClustersà la section Référence des AWS CLI commandes.

`describe-db-engine-versions`

L'exemple de code suivant montre comment utiliser`describe-db-engine-versions`.

AWS CLI

Pour décrire les versions du moteur de base de données pour le moteur de base de données MySQL

L'`describe-db-engine-versions`exemple suivant affiche des détails sur chacune des versions du moteur de base de données pour le moteur de base de données spécifié.

```
aws rds describe-db-engine-versions \
    --engine mysql
```

Sortie :

```
{
    "DBEngineVersions": [
        {
            "Engine": "mysql",
            "EngineVersion": "5.5.46",
            "DBParameterGroupFamily": "mysql5.5",
            "DBEngineDescription": "MySQL Community Edition",
            "DBEngineVersionDescription": "MySQL 5.5.46",
            "ValidUpgradeTarget": [
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.53",
                    "Description": "MySQL 5.5.53",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.54",
                    "Description": "MySQL 5.5.54",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.57",
                    "Description": "MySQL 5.5.57",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
```

```
                ...some output truncated...
            ]
        }
```

Pour plus d'informations, consultez Qu'est-ce qu'Amazon Relational Database Service (Amazon RDS) ? dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Décrire DBEngine les versions dans AWS CLI la référence des commandes.

`describe-db-instance-automated-backups`

L'exemple de code suivant montre comment utiliser`describe-db-instance-automated-backups`.

AWS CLI

Pour décrire les sauvegardes automatisées d'une instance de base de données

L'`describe-db-instance-automated-backups`exemple suivant affiche des détails sur les sauvegardes automatisées pour l'instance de base de données spécifiée. Les détails incluent les sauvegardes automatisées répliquées dans d'autres AWS régions.

```
aws rds describe-db-instance-automated-backups \
    --db-instance-identifier new-orcl-db
```

Sortie :

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
            "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
            "Region": "us-east-1",
            "DBInstanceIdentifier": "new-orcl-db",
            "RestoreWindow": {
                "EarliestTime": "2020-12-07T21:05:20.939Z",
                "LatestTime": "2020-12-07T21:05:20.939Z"
            },
            "AllocatedStorage": 20,
            "Status": "replicating",
```

```
            "Port": 1521,
            "InstanceCreateTime": "2020-12-04T15:28:31Z",
            "MasterUsername": "admin",
            "Engine": "oracle-se2",
            "EngineVersion": "12.1.0.2.v21",
            "LicenseModel": "bring-your-own-license",
            "OptionGroupName": "default:oracle-se2-12-1",
            "Encrypted": false,
            "StorageType": "gp2",
            "IAMDatabaseAuthenticationEnabled": false,
            "BackupRetentionPeriod": 14,
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-
west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
        }
    ]
}
```

Pour plus d'informations, consultez la section Recherche d'informations sur les sauvegardes répliquées dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeDbInstanceAutomatedBackupsà la section Référence des AWS CLI commandes.

`describe-db-instances`

L'exemple de code suivant montre comment utiliser`describe-db-instances`.

AWS CLI

Pour décrire une instance de base de données

L'`describe-db-instances`exemple suivant récupère les détails de l'instance de base de données spécifiée.

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

Sortie :

```
{
    "DBInstances": [
        {
```

```
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-east-1.rds.amazonaws.com",
                "Port": 3306,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            ...some output truncated...
        }
    ]
}
```

- Pour plus de détails sur l'API, voir Description DBInstances dans le manuel de référence des AWS CLI commandes.

`describe-db-log-files`

L'exemple de code suivant montre comment utiliser`describe-db-log-files`.

AWS CLI

Pour décrire les fichiers journaux d'une instance de base de données

L'`describe-db-log-files`exemple suivant récupère des détails sur les fichiers journaux de l'instance de base de données spécifiée.

```
aws rds describe-db-log-files -\
    -db-instance-identifier test-instance
```

Sortie :

```
{
    "DescribeDBLogFiles": [
        {
            "Size": 0,
            "LastWritten": 1533060000000,
            "LogFileName": "error/mysql-error-running.log"
        },
        {
```

```
            "Size": 2683,
            "LastWritten": 1532994300000,
            "LogFileName": "error/mysql-error-running.log.0"
        },
        {
            "Size": 107,
            "LastWritten": 1533057300000,
            "LogFileName": "error/mysql-error-running.log.18"
        },
        {
            "Size": 13105,
            "LastWritten": 1532991000000,
            "LogFileName": "error/mysql-error-running.log.23"
        },
        {
            "Size": 0,
            "LastWritten": 1533061200000,
            "LogFileName": "error/mysql-error.log"
        },
        {
            "Size": 3519,
            "LastWritten": 1532989252000,
            "LogFileName": "mysqlUpgrade"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDbLogFilesà la section Référence des AWS CLI commandes.

`describe-db-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-db-parameter-groups`.

AWS CLI

Pour décrire votre groupe de paramètres de base de données

L'`describe-db-parameter-groups`exemple suivant permet de récupérer des informations sur vos groupes de paramètres de base de données.

```
aws rds describe-db-parameter-groups
```

Sortie :

```
{
    "DBParameterGroups": [
        {
            "DBParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default parameter group for aurora-mysql5.7",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
        },
        {
            "DBParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default parameter group for aurora-postgresql9.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
        },
        {
            "DBParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default parameter group for aurora5.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
        },
        {
            "DBParameterGroupName": "default.mariadb10.1",
            "DBParameterGroupFamily": "mariadb10.1",
            "Description": "Default parameter group for mariadb10.1",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
        },
        ...some output truncated...
    ]
}
```

Pour de plus amples informations, veuillez consulter Utilisation des groupes de paramètres de base de données dans le Guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, consultez la section Décrire DBParameter les groupes dans AWS CLI la référence des commandes.

`describe-db-parameters`

L'exemple de code suivant montre comment utiliser`describe-db-parameters`.

AWS CLI

Pour décrire les paramètres d'un groupe de paramètres de base de données

L'`describe-db-parameters`exemple suivant récupère les détails du groupe de paramètres de base de données spécifié.

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

Sortie :

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "auto_generate_certs",
            "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        ...some output truncated...
    ]
}
```

Pour de plus amples informations, veuillez consulter Utilisation des groupes de paramètres de base de données dans le Guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Description DBParameters dans le manuel de référence des AWS CLI commandes.

`describe-db-proxies`

L'exemple de code suivant montre comment utiliser`describe-db-proxies`.

AWS CLI

Pour décrire un proxy de base de données pour une base de données RDS

L'`describe-db-proxies`exemple suivant renvoie des informations sur les proxys de base de données.

```
aws rds describe-db-proxies
```

Sortie :

```
{
    "DBProxies": [
        {
            "DBProxyName": "proxyExample1",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
            "Status": "available",
            "EngineFamily": "PostgreSQL",
            "VpcId": "vpc-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                    "Description": "proxydescription1"
                    "AuthScheme": "SECRETS",
                    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
```

```
                    "IAMAuth": "DISABLED"
                }
            ]",
            "RoleArn": "arn:aws:iam::12345678912??:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample1.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
        },
        {
            "DBProxyName": "proxyExample2",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-1234a12b23456c1ab",
            "Status": "available",
            "EngineFamily": "PostgreSQL",
            "VpcId": "sg-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                    "Description": "proxydescription2"
                    "AuthScheme": "SECRETS",
                    "SecretArn": "aarn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
                    "IAMAuth": "DISABLED"
                }
            ]",
            "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample2.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
            "CreatedDate": "2022-01-05T16:19:33.452000+00:00",
            "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
        }
```

```
    ]
}
```

Pour plus d'informations, consultez les sections Affichage d'un proxy RDS dans le guide de l'utilisateur Amazon RDS et Affichage d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbProxiesà la section Référence des AWS CLI commandes.

`describe-db-proxy-endpoints`

L'exemple de code suivant montre comment utiliser`describe-db-proxy-endpoints`.

AWS CLI

Pour décrire les points de terminaison d'un proxy de base de données

L'`describe-db-proxy-endpoints`exemple suivant renvoie des informations sur les points de terminaison du proxy de base de données.

```
aws rds describe-db-proxy-endpoints
```

Sortie :

```
{
    "DBProxyEndpoints": [
        {
            "DBProxyEndpointName": "proxyEndpoint1",
            "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
            "DBProxyName": "proxyExample",
            "Status": "available",
            "VpcId": "vpc-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
```

```
            "Endpoint": "proxyEndpoint1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "TargetRole": "READ_WRITE",
            "IsDefault": false
        },
        {
            "DBProxyEndpointName": "proxyEndpoint2",
            "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-4567a01b12345c0ab",
            "DBProxyName": "proxyExample2",
            "Status": "available",
            "VpcId": "vpc1234567",
            "VpcSecurityGroupIds": [
                "sg-5678"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Endpoint": "proxyEndpoint2.endpoint.proxy-cd1ef2klmnop.us-
east-1.rds.amazonaws.com",
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "TargetRole": "READ_WRITE",
            "IsDefault": false
        }
    ]
}
```

Pour plus d'informations, consultez les sections Affichage d'un point de terminaison proxy dans le guide de l'utilisateur Amazon RDS et Création d'un point de terminaison proxy dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbProxyEndpointsà la section Référence des AWS CLI commandes.

`describe-db-proxy-target-groups`

L'exemple de code suivant montre comment utiliser`describe-db-proxy-target-groups`.

AWS CLI

Pour décrire les points de terminaison d'un proxy de base de données

L'`describe-db-proxy-target-groups`exemple suivant renvoie des informations sur les groupes cibles de proxy de base de données.

```
aws rds describe-db-proxy-target-groups \
    --db-proxy-name proxyExample
```

Sortie :

```
{
"TargetGroups":
    {
        "DBProxyName": "proxyExample",
        "TargetGroupName": "default",
        "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
        "IsDefault": true,
        "Status": "available",
        "ConnectionPoolConfig": {
            "MaxConnectionsPercent": 100,
            "MaxIdleConnectionsPercent": 50,
            "ConnectionBorrowTimeout": 120,
            "SessionPinningFilters": []
        },
        "CreatedDate": "2023-05-02T18:41:19.495000+00:00",
        "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

Pour plus d'informations, consultez les sections Affichage d'un proxy RDS dans le guide de l'utilisateur Amazon RDS et Affichage d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbProxyTargetGroupsà la section Référence des AWS CLI commandes.

`describe-db-proxy-targets`

L'exemple de code suivant montre comment utiliser`describe-db-proxy-targets`.

AWS CLI

Pour décrire les cibles proxy de base de données

L'`describe-db-proxy-targets`exemple suivant renvoie des informations sur les cibles proxy de base de données.

```
aws rds describe-db-proxy-targets \
    --db-proxy-name proxyExample
```

Sortie :

```
{
    "Targets": [
        {
            "Endpoint": "database1.ab0cd1efghij.us-east-1.rds.amazonaws.com",
            "TrackedClusterId": "database1",
            "RdsResourceId": "database1-instance-1",
            "Port": 3306,
            "Type": "RDS_INSTANCE",
            "Role": "READ_WRITE",
            "TargetHealth": {
                "State": "UNAVAILABLE",
                "Reason": "PENDING_PROXY_CAPACITY",
                "Description": "DBProxy Target is waiting for proxy to scale to
 desired capacity"
            }
        }
    ]
}
```

Pour plus d'informations, consultez les sections Affichage d'un proxy RDS dans le guide de l'utilisateur Amazon RDS et Affichage d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbProxyTargetsà la section Référence des AWS CLI commandes.

`describe-db-recommendations`

L'exemple de code suivant montre comment utiliser`describe-db-recommendations`.

AWS CLI

Exemple 1 : pour répertorier toutes les recommandations de base de données

L'`describe-db-recommendations`exemple suivant répertorie toutes les recommandations de base de données de votre AWS compte.

```
aws rds describe-db-recommendations
```

Sortie :

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::old_minor_version",
            "Severity": "informational",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "**[resource-name]** is not running the latest minor DB
 engine version",
            "Recommendation": "Upgrade to latest engine version",
            "Description": "Your database resources aren't running the latest minor
 DB engine version. The latest minor version contains the latest security fixes and
 other improvements.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Operation": "modifyDbInstance",
                    "Parameters": [
                        {
                            "Key": "EngineVersion",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "DBInstanceIdentifier",
                            "Value": "database-1"
                        }
                    ],
                    "ApplyModes": [
                        "immediately",
                        "next-maintenance-window"
                    ],
                    "Status": "ready",
                    "ContextAttributes": [
```

```
                        {
                            "Key": "Recommended value",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "Current engine version",
                            "Value": "5.7.42"
                        }
                    ]
                }
            ],
            "Category": "security",
            "Source": "RDS",
            "TypeDetection": "**[resource-count] resources** are not running the
 latest minor DB engine version",
            "TypeRecommendation": "Upgrade to latest engine version",
            "Impact": "Reduced database performance and data security at risk",
            "AdditionalInfo": "We recommend that you maintain your database with the
 latest DB engine minor version as this version includes the latest security and
 functionality fixes. The DB engine minor version upgrades contain only the changes
 which are backward-compatible with earlier minor versions of the same major version
 of the DB engine.",
            "Links": [
                {
                    "Text": "Upgrading an RDS DB instance engine version",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon Aurora",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon RDS",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
                }
            ]
        }
    ]
```

```
}
```

Pour plus d'informations, consultez Consulter les recommandations Amazon RDS et y répondre dans le guide de l'utilisateur Amazon RDS et Afficher et répondre aux recommandations Amazon RDS dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : Pour répertorier les recommandations de base de données les plus sévères

L'`describe-db-recommendations`exemple suivant répertorie les recommandations de base de données les plus sévères pour votre AWS compte.

```
aws rds describe-db-recommendations \
    --filters Name=severity,Values=high
```

Sortie :

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::rds_extended_support",
            "Severity": "high",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.392000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "Your databases will be auto-enrolled to RDS Extended
 Support on February 29",
            "Recommendation": "Upgrade your major version before February 29, 2024
 to avoid additional charges",
            "Description": "Your PostgreSQL 11 and MySQL 5.7 databases will be
 automatically enrolled into RDS Extended Support on February 29, 2024. To avoid
 the increase in charges due to RDS Extended Support, we recommend upgrading your
 databases to a newer major engine version before February 29, 2024.\nTo learn more
 about the RDS Extended Support pricing, refer to the pricing page.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Parameters": [],
                    "ApplyModes": [
                        "manual"
                    ],
                    "Status": "ready",
```

```
                    "ContextAttributes": []
                }
            ],
            "Category": "cost optimization",
            "Source": "RDS",
            "TypeDetection": "Your database will be auto-enrolled to RDS Extended
 Support on February 29",
            "TypeRecommendation": "Upgrade your major version before February 29,
 2024 to avoid additional charges",
            "Impact": "Increase in charges due to RDS Extended Support",
            "AdditionalInfo": "With Amazon RDS Extended Support, you can continue
 running your database on a major engine version past the RDS end of standard
 support date for an additional cost. This paid feature gives you more time to
 upgrade to a supported major engine version.\nDuring Extended Support, Amazon RDS
 will supply critical CVE patches and bug fixes.",
            "Links": [
                {
                    "Text": "Amazon RDS Extended Support pricing for RDS for MySQL",
                    "Url": "https://aws.amazon.com/rds/mysql/pricing/"
                },
                {
                    "Text": "Amazon RDS Extended Support for RDS for MySQL and
 PostgreSQL databases",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
extended-support.html"
                },
                {
                    "Text": "Amazon RDS Extended Support pricing for Amazon Aurora
 PostgreSQL",
                    "Url": "https://aws.amazon.com/rds/aurora/pricing/"
                },
                {
                    "Text": "Amazon RDS Extended Support for Aurora PostgreSQL
 databases",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/extended-support.html"
                },
                {
                    "Text": "Amazon RDS Extended Support pricing for RDS for
 PostgreSQL",
                    "Url": "https://aws.amazon.com/rds/postgresql/pricing/"
                }
            ]
        }
```

```
    ]
}
```

Pour plus d'informations, consultez Consulter les recommandations Amazon RDS et y répondre dans le guide de l'utilisateur Amazon RDS et Afficher et répondre aux recommandations Amazon RDS dans le guide de l'utilisateur Amazon Aurora.

Exemple 3 : pour répertorier les recommandations de base de données pour une instance de base de données spécifiée

L'`describe-db-recommendations`exemple suivant répertorie toutes les recommandations de base de données pour une instance de base de données spécifiée.

```
aws rds describe-db-recommendations \
    --filters Name=dbi-resource-id,Values=database-1
```

Sortie :

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::old_minor_version",
            "Severity": "informational",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "**[resource-name]** is not running the latest minor DB
 engine version",
            "Recommendation": "Upgrade to latest engine version",
            "Description": "Your database resources aren't running the latest minor
 DB engine version. The latest minor version contains the latest security fixes and
 other improvements.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Operation": "modifyDbInstance",
                    "Parameters": [
                        {
                            "Key": "EngineVersion",
                            "Value": "5.7.44"
```

```
                        },
                        {
                            "Key": "DBInstanceIdentifier",
                            "Value": "database-1"
                        }
                    ],
                    "ApplyModes": [
                        "immediately",
                        "next-maintenance-window"
                    ],
                    "Status": "ready",
                    "ContextAttributes": [
                        {
                            "Key": "Recommended value",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "Current engine version",
                            "Value": "5.7.42"
                        }
                    ]
                }
            ],
            "Category": "security",
            "Source": "RDS",
            "TypeDetection": "**[resource-count] resources** are not running the
 latest minor DB engine version",
            "TypeRecommendation": "Upgrade to latest engine version",
            "Impact": "Reduced database performance and data security at risk",
            "AdditionalInfo": "We recommend that you maintain your database with the
 latest DB engine minor version as this version includes the latest security and
 functionality fixes. The DB engine minor version upgrades contain only the changes
 which are backward-compatible with earlier minor versions of the same major version
 of the DB engine.",
            "Links": [
                {
                    "Text": "Upgrading an RDS DB instance engine version",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon Aurora",
```

```
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon RDS",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Consulter les recommandations Amazon RDS et y répondre dans le guide de l'utilisateur Amazon RDS et Afficher et répondre aux recommandations Amazon RDS dans le guide de l'utilisateur Amazon Aurora.

Exemple 4 : pour répertorier toutes les recommandations de base de données actives

L'`describe-db-recommendations`exemple suivant répertorie toutes les recommandations de base de données actives dans votre AWS compte.

```
aws rds describe-db-recommendations \
    --filters Name=status,Values=active
```

Sortie :

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::old_minor_version",
            "Severity": "informational",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "**[resource-name]** is not running the latest minor DB
 engine version",
            "Recommendation": "Upgrade to latest engine version",
```

```
            "Description": "Your database resources aren't running the latest minor
DB engine version. The latest minor version contains the latest security fixes and
other improvements.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Operation": "modifyDbInstance",
                    "Parameters": [
                        {
                            "Key": "EngineVersion",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "DBInstanceIdentifier",
                            "Value": "database-1"
                        }
                    ],
                    "ApplyModes": [
                        "immediately",
                        "next-maintenance-window"
                    ],
                    "Status": "ready",
                    "ContextAttributes": [
                        {
                            "Key": "Recommended value",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "Current engine version",
                            "Value": "5.7.42"
                        }
                    ]
                }
            ],
            "Category": "security",
            "Source": "RDS",
            "TypeDetection": "**[resource-count] resources** are not running the
latest minor DB engine version",
            "TypeRecommendation": "Upgrade to latest engine version",
            "Impact": "Reduced database performance and data security at risk",
            "AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
```

```
 which are backward-compatible with earlier minor versions of the same major version
 of the DB engine.",
            "Links": [
                {
                    "Text": "Upgrading an RDS DB instance engine version",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon Aurora",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon RDS",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Consulter les recommandations Amazon RDS et y répondre dans le guide de l'utilisateur Amazon RDS et Afficher et répondre aux recommandations Amazon RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbRecommendationsà la section Référence des AWS CLI commandes.

`describe-db-security-groups`

L'exemple de code suivant montre comment utiliser`describe-db-security-groups`.

AWS CLI

Pour répertorier les groupes de sécurité de base de données

L'`describe-db-security-groups`exemple suivant répertorie les groupes de sécurité de base de données.

```
aws rds describe-db-security-groups
```

Sortie :

```
{
    "DBSecurityGroups": [
        {
            "OwnerId": "123456789012",
            "DBSecurityGroupName": "default",
            "DBSecurityGroupDescription": "default",
            "EC2SecurityGroups": [],
            "IPRanges": [],
            "DBSecurityGroupArn": "arn:aws:rds:us-
west-1:111122223333:secgrp:default"
        },
        {
            "OwnerId": "123456789012",
            "DBSecurityGroupName": "mysecgroup",
            "DBSecurityGroupDescription": "My Test Security Group",
            "VpcId": "vpc-1234567f",
            "EC2SecurityGroups": [],
            "IPRanges": [],
            "DBSecurityGroupArn": "arn:aws:rds:us-
west-1:111122223333:secgrp:mysecgroup"
        }
    ]
}
```

Pour plus d'informations, consultez la section Liste des groupes de sécurité de base de données disponibles dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeDbSecurityGroupsà la section Référence des AWS CLI commandes.

`describe-db-shard-groups`

L'exemple de code suivant montre comment utiliser`describe-db-shard-groups`.

AWS CLI

Exemple 1 : pour décrire les groupes de partitions de base de données

L'`describe-db-shard-groups`exemple suivant récupère les détails de vos groupes de partitions de base de données.

```
aws rds describe-db-shard-groups
```

Sortie :

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Aurora DB Clusters dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeDbShardGroupsà la section Référence des AWS CLI commandes.

`describe-db-snapshot-attributes`

L'exemple de code suivant montre comment utiliser`describe-db-snapshot-attributes`.

AWS CLI

Pour décrire les noms et les valeurs des attributs d'un instantané de base de données

L'`describe-db-snapshot-attributes`exemple suivant décrit les noms et les valeurs des attributs d'un instantané de base de données.

```
aws rds describe-db-snapshot-attributes \
    --db-snapshot-identifier mydbsnapshot
```

Sortie :

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012",
                    "210987654321"
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Partage d'un instantané de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeDbSnapshotAttributesà la section Référence des AWS CLI commandes.

`describe-db-snapshots`

L'exemple de code suivant montre comment utiliser`describe-db-snapshots`.

AWS CLI

Exemple 1 : pour décrire un instantané de base de données pour une instance de base de données

L'`describe-db-snapshots`exemple suivant récupère les détails d'un instantané de base de données pour une instance de base de données.

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

Sortie :

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "mydbsnapshot",
            "DBInstanceIdentifier": "mysqldb",
            "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
            "Engine": "mysql",
            "AllocatedStorage": 20,
            "Status": "available",
            "Port": 3306,
            "AvailabilityZone": "us-east-1f",
            "VpcId": "vpc-6594f31c",
            "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
            "MasterUsername": "mysqladmin",
            "EngineVersion": "5.6.37",
            "LicenseModel": "general-public-license",
            "SnapshotType": "manual",
            "OptionGroupName": "default:mysql-5-6",
            "PercentProgress": 100,
            "StorageType": "gp2",
            "Encrypted": false,
            "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
            "IAMDatabaseAuthenticationEnabled": false,
            "ProcessorFeatures": [],
            "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création d'un instantané de base de données dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour trouver le nombre de clichés pris manuellement

L'`describe-db-snapshots`exemple suivant utilise l'`length`opérateur dans l'`--query`option pour renvoyer le nombre de clichés manuels qui ont été pris dans une AWS région donnée.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

Sortie :

```
35
```

Pour plus d'informations, consultez la section Création d'un instantané de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Description DBSnapshots dans le manuel de référence des AWS CLI commandes.

`describe-db-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-db-subnet-groups`.

AWS CLI

Pour décrire un groupe de sous-réseaux de base de données

L'`describe-db-subnet-groups`exemple suivant récupère les détails du groupe de sous-réseaux de base de données spécifié.

```
aws rds describe-db-subnet-groups
```

Sortie :

```
{
    "DBSubnetGroups": [
        {
            "DBSubnetGroupName": "mydbsubnetgroup",
```

```
            "DBSubnetGroupDescription": "My DB Subnet Group",
            "VpcId": "vpc-971c12ee",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-d8c8e7f4",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-718fdc7d",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-cbc8e7e7",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-0ccde220",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                }
            ],
            "DBSubnetGroupArn": "arn:aws:rds:us-
east-1:123456789012:subgrp:mydbsubnetgroup"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Virtual Private Cloud VPCs et Amazon RDS dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeDbSubnetGroupsà la section Référence des AWS CLI commandes.

`describe-engine-default-cluster-parameters`

L'exemple de code suivant montre comment utiliser`describe-engine-default-cluster-parameters`.

AWS CLI

Pour décrire le moteur par défaut et les informations sur les paramètres système du moteur de base de données Aurora

L'`describe-engine-default-cluster-parameters`exemple suivant récupère les détails du moteur par défaut et les informations sur les paramètres système pour les clusters de base de données Aurora compatibles avec MySQL 5.7.

```
aws rds describe-engine-default-cluster-parameters \
    --db-parameter-group-family aurora-mysql5.7
```

Sortie :

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "aurora_load_from_s3_role",
                "Description": "IAM role ARN used to load data from AWS S3",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "string",
                "IsModifiable": true,
                "SupportedEngineModes": [
                    "provisioned"
                ]
            },
            ...some output truncated...
        ]
    }
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeEngineDefaultClusterParametersà la section Référence des AWS CLI commandes.

`describe-engine-default-parameters`

L'exemple de code suivant montre comment utiliser`describe-engine-default-parameters`.

AWS CLI

Pour décrire le moteur par défaut et les informations sur les paramètres système du moteur de base de données

L'`describe-engine-default-parameters`exemple suivant récupère les détails du moteur par défaut et les informations des paramètres système pour les instances de base de données MySQL 5.7.

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family mysql5.7
```

Sortie :

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "allow-suspicious-udfs",
                "Description": "Controls whether user-defined functions that have
 only an xxx symbol for the main function can be loaded",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
            ...some output truncated...
        ]
    }
}
```

Pour de plus amples informations, veuillez consulter Utilisation des groupes de paramètres de base de données dans le Guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeEngineDefaultParametersà la section Référence des AWS CLI commandes.

`describe-event-categories`

L'exemple de code suivant montre comment utiliser`describe-event-categories`.

AWS CLI

Pour décrire les catégories d'événements

L'`describe-event-categories`exemple suivant permet de récupérer des informations sur les catégories d'événements pour toutes les sources d'événements disponibles.

```
aws rds describe-event-categories
```

Sortie :

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-instance",
            "EventCategories": [
                "deletion",
                "read replica",
                "failover",
                "restoration",
                "maintenance",
                "low storage",
                "configuration change",
                "backup",
                "creation",
                "availability",
                "recovery",
                "failure",
                "backtrack",
                "notification"
            ]
        },
        {
            "SourceType": "db-security-group",
            "EventCategories": [
```

```
                "configuration change",
                "failure"
            ]
        },
        {
            "SourceType": "db-parameter-group",
            "EventCategories": [
                "configuration change"
            ]
        },
        {
            "SourceType": "db-snapshot",
            "EventCategories": [
                "deletion",
                "creation",
                "restoration",
                "notification"
            ]
        },
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "failover",
                "failure",
                "notification"
            ]
        },
        {
            "SourceType": "db-cluster-snapshot",
            "EventCategories": [
                "backup"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEventCategoriesà la section Référence des AWS CLI commandes.

`describe-event-subscriptions`

L'exemple de code suivant montre comment utiliser`describe-event-subscriptions`.

AWS CLI

Pour décrire les abonnements aux événements

Cet exemple décrit tous les abonnements aux événements Amazon RDS pour le AWS compte courant.

```
aws rds describe-event-subscriptions
```

Sortie :

```
{
    "EventSubscriptionsList": [
        {
            "EventCategoriesList": [
                "backup",
                "recovery"
            ],
            "Enabled": true,
            "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-
instance-events",
            "Status": "creating",
            "SourceType": "db-instance",
            "CustomerAwsId": "123456789012",
            "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
            "CustSubscriptionId": "my-instance-events",
            "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events"
        },
        ...some output truncated...
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEventSubscriptionsà la section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Pour décrire les événements

L'`describe-events`exemple suivant récupère les détails des événements survenus pour l'instance de base de données spécifiée.

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance
```

Sortie :

```
{
    "Events": [
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Backing up DB instance",
            "Date": "2018-07-31T23:09:23.983Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        },
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Finished DB Instance backup",
            "Date": "2018-07-31T23:15:13.049Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEventsà la section Référence des AWS CLI commandes.

`describe-export-tasks`

L'exemple de code suivant montre comment utiliser`describe-export-tasks`.

AWS CLI

Pour décrire les tâches d'exportation de snapshots

L'`describe-export-tasks`exemple suivant renvoie des informations sur les exportations de snapshots vers Amazon S3.

```
aws rds describe-export-tasks
```

Sortie :

```
{
    "ExportTasks": [
        {
            "ExportTaskIdentifier": "test-snapshot-export",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:test-
snapshot",
            "SnapshotTime": "2020-03-02T18:26:28.163Z",
            "TaskStartTime": "2020-03-02T18:57:56.896Z",
            "TaskEndTime": "2020-03-02T19:10:31.985Z",
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Prefix": "",
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
            "Status": "COMPLETE",
            "PercentProgress": 100,
            "TotalExtractedDataInGB": 0
        },
        {
            "ExportTaskIdentifier": "my-s3-export",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
test",
            "SnapshotTime": "2020-03-27T20:48:42.023Z",
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Prefix": "",
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
            "Status": "STARTING",
            "PercentProgress": 0,
            "TotalExtractedDataInGB": 0
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Surveillance des exportations de snapshots dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeExportTasksà la section Référence des AWS CLI commandes.

`describe-global-clusters`

L'exemple de code suivant montre comment utiliser`describe-global-clusters`.

AWS CLI

Pour décrire les clusters de bases de données globaux

L'`describe-global-clusters`exemple suivant répertorie les clusters de base de données globaux Aurora dans la AWS région actuelle.

```
aws rds describe-global-clusters
```

Sortie :

```
{
    "GlobalClusters": [
        {
            "GlobalClusterIdentifier": "myglobalcluster",
            "GlobalClusterResourceId": "cluster-f5982077e3b5aabb",
            "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
            "Status": "available",
            "Engine": "aurora-mysql",
            "EngineVersion": "5.7.mysql_aurora.2.07.2",
            "StorageEncrypted": false,
            "DeletionProtection": false,
            "GlobalClusterMembers": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion d'une base de données globale Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous DescribeGlobalClustersà la section Référence des AWS CLI commandes.

`describe-option-group-options`

L'exemple de code suivant montre comment utiliser`describe-option-group-options`.

AWS CLI

Pour décrire toutes les options disponibles

L'`describe-option-group-options`exemple suivant répertorie deux options pour une instance Oracle Database 19c.

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19 \
    --max-items 2
```

Sortie :

```
{
    "OptionGroupOptions": [
        {
            "Name": "APEX",
            "Description": "Oracle Application Express Runtime Environment",
            "EngineName": "oracle-ee",
            "MajorEngineVersion": "19",
            "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
            "PortRequired": false,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "SupportsOptionVersionDowngrade": false,
            "OptionGroupOptionSettings": [],
            "OptionGroupOptionVersions": [
                {
```

```
                    "Version": "19.1.v1",
                    "IsDefault": true
                },
                {
                    "Version": "19.2.v1",
                    "IsDefault": false
                }
            ]
        },
        {
            "Name": "APEX-DEV",
            "Description": "Oracle Application Express Development Environment",
            "EngineName": "oracle-ee",
            "MajorEngineVersion": "19",
            "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
            "PortRequired": false,
            "OptionsDependedOn": [
                "APEX"
            ],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": []
        }
    ],
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```


Pour plus d'informations, consultez la section Liste des options et des paramètres d'options pour un groupe d'options dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeOptionGroupOptionsà la section Référence des AWS CLI commandes.

`describe-option-groups`

L'exemple de code suivant montre comment utiliser`describe-option-groups`.

AWS CLI

Pour décrire les groupes d'options disponibles

L'`describe-option-groups`exemple suivant répertorie les groupes d'options pour une instance Oracle Database 19c.

```
aws rds describe-option-groups \
    --engine-name oracle-ee \
    --major-engine-version 19
```

Sortie :

```
{
    "OptionGroupsList": [
        {
            "OptionGroupName": "default:oracle-ee-19",
            "OptionGroupDescription": "Default option group for oracle-ee 19",
            "EngineName": "oracle-ee",
            "MajorEngineVersion": "19",
            "Options": [],
            "AllowsVpcAndNonVpcInstanceMemberships": true,
            "OptionGroupArn": "arn:aws:rds:us-west-1:111122223333:og:default:oracle-
ee-19"
        }
    ]
}
```

Pour plus d'informations, consultez la section Liste des options et des paramètres d'options pour un groupe d'options dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeOptionGroupsà la section Référence des AWS CLI commandes.

`describe-orderable-db-instance-options`

L'exemple de code suivant montre comment utiliser`describe-orderable-db-instance-options`.

AWS CLI

Pour décrire les options d'instance de base de données pouvant être commandées

L'`describe-orderable-db-instance-options`exemple suivant récupère des détails sur les options commandables pour une instance de base de données exécutant le moteur de base de données MySQL.

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

Sortie :

```
{
    "OrderableDBInstanceOptions": [
        {
            "MinStorageSize": 5,
            "ReadReplicaCapable": true,
            "MaxStorageSize": 6144,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "SupportsIops": false,
            "AvailableProcessorFeatures": [],
            "MultiAZCapable": true,
            "DBInstanceClass": "db.m1.large",
            "Vpc": true,
            "StorageType": "gp2",
            "LicenseModel": "general-public-license",
            "EngineVersion": "5.5.46",
            "SupportsStorageEncryption": false,
            "SupportsEnhancedMonitoring": true,
            "Engine": "mysql",
            "SupportsIAMDatabaseAuthentication": false,
            "SupportsPerformanceInsights": false
        }
    ]
    ...some output truncated...
}
```

- Pour plus de détails sur l'API, consultez la section DescribeOrderableDBInstanceOptions du manuel de référence des AWS CLI commandes.

`describe-pending-maintenance-actions`

L'exemple de code suivant montre comment utiliser`describe-pending-maintenance-actions`.

AWS CLI

Pour répertorier les ressources dont au moins une action de maintenance est en attente

L'`describe-pending-maintenance-actions`exemple suivant répertorie l'action de maintenance en attente pour une instance de base de données.

```
aws rds describe-pending-maintenance-actions
```

Sortie :

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
west-2:123456789012:cluster:global-db1-cl1",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "system-update",
                    "Description": "Upgrade to Aurora PostgreSQL 2.4.2"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Maintenance d'une instance de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribePendingMaintenanceActionsà la section Référence des AWS CLI commandes.

`describe-reserved-db-instances-offerings`

L'exemple de code suivant montre comment utiliser`describe-reserved-db-instances-offerings`.

AWS CLI

Pour décrire les offres d'instances de base de données réservées

L'`describe-reserved-db-instances-offerings`exemple suivant récupère des détails sur les options d'instance de base de données réservées pour`oracle`.

```
aws rds describe-reserved-db-instances-offerings \
    --product-description oracle
```

Sortie :

```
{
    "ReservedDBInstancesOfferings": [
        {
            "CurrencyCode": "USD",
            "UsagePrice": 0.0,
            "ProductDescription": "oracle-se2(li)",
            "ReservedDBInstancesOfferingId": "005bdee3-9ef4-4182-aa0c-58ef7cb6c2f8",
            "MultiAZ": true,
            "DBInstanceClass": "db.m4.xlarge",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.594,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "FixedPrice": 4089.0,
            "Duration": 31536000
        },
    ...some output truncated...
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeReservedDbInstancesOfferingsà la section Référence des AWS CLI commandes.

`describe-reserved-db-instances`

L'exemple de code suivant montre comment utiliser`describe-reserved-db-instances`.

AWS CLI

Pour décrire les instances de base de données réservées

L'`describe-reserved-db-instances`exemple suivant récupère les détails de toutes les instances de base de données réservées dans le AWS compte courant.

```
aws rds describe-reserved-db-instances
```

Sortie :

```
{
    "ReservedDBInstances": [
        {
            "ReservedDBInstanceId": "myreservedinstance",
            "ReservedDBInstancesOfferingId": "12ab34cd-59af-4b2c-a660-1abcdef23456",
            "DBInstanceClass": "db.t3.micro",
            "StartTime": "2020-06-01T13:44:21.436Z",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CurrencyCode": "USD",
            "DBInstanceCount": 1,
            "ProductDescription": "sqlserver-ex(li)",
            "OfferingType": "No Upfront",
            "MultiAZ": false,
            "State": "payment-pending",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.014,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ReservedDBInstanceArn": "arn:aws:rds:us-
west-2:123456789012:ri:myreservedinstance",
            "LeaseId": "a1b2c3d4-6b69-4a59-be89-5e11aa446666"
        }
    ]
}
```

Pour plus d'informations, consultez la section Instances de base de données réservées pour Amazon RDS dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeReservedDbInstancesà la section Référence des AWS CLI commandes.

`describe-source-regions`

L'exemple de code suivant montre comment utiliser`describe-source-regions`.

AWS CLI

Pour décrire les régions source

L'`describe-source-regions`exemple suivant permet de récupérer des informations sur toutes les AWS régions sources. Cela montre également que les sauvegardes automatisées ne peuvent être répliquées que depuis l'ouest des États-Unis (Oregon) vers la AWS région de destination, l'est des États-Unis (Virginie du Nord).

```
aws rds describe-source-regions \
    --region us-east-1
```

Sortie :

```
{
    "SourceRegions": [
        {
            "RegionName": "af-south-1",
            "Endpoint": "https://rds.af-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ap-east-1",
            "Endpoint": "https://rds.ap-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ap-northeast-1",
            "Endpoint": "https://rds.ap-northeast-1.amazonaws.com",
            "Status": "available",
```

```
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-northeast-2",
            "Endpoint": "https://rds.ap-northeast-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-northeast-3",
            "Endpoint": "https://rds.ap-northeast-3.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ap-south-1",
            "Endpoint": "https://rds.ap-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-southeast-1",
            "Endpoint": "https://rds.ap-southeast-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-southeast-2",
            "Endpoint": "https://rds.ap-southeast-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-southeast-3",
            "Endpoint": "https://rds.ap-southeast-3.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ca-central-1",
            "Endpoint": "https://rds.ca-central-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
```

```
        {
            "RegionName": "eu-north-1",
            "Endpoint": "https://rds.eu-north-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "eu-south-1",
            "Endpoint": "https://rds.eu-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "eu-west-1",
            "Endpoint": "https://rds.eu-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "eu-west-2",
            "Endpoint": "https://rds.eu-west-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "eu-west-3",
            "Endpoint": "https://rds.eu-west-3.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "me-central-1",
             "Endpoint": "https://rds.me-central-1.amazonaws.com",
             "Status": "available",
             "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "me-south-1",
            "Endpoint": "https://rds.me-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "sa-east-1",
```

```
            "Endpoint": "https://rds.sa-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-east-2",
            "Endpoint": "https://rds.us-east-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-west-1",
            "Endpoint": "https://rds.us-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-west-2",
           "Endpoint": "https://rds.us-west-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        }
    ]
}
```

Pour plus d'informations, consultez la section Recherche d'informations sur les sauvegardes répliquées dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous DescribeSourceRegionsà la section Référence des AWS CLI commandes.

`describe-valid-db-instance-modifications`

L'exemple de code suivant montre comment utiliser`describe-valid-db-instance-modifications`.

AWS CLI

Pour décrire les modifications valides pour une instance de base de données

L'`describe-valid-db-instance-modifications`exemple suivant récupère des détails sur les modifications valides pour l'instance de base de données spécifiée.

```
aws rds describe-valid-db-instance-modifications \
    --db-instance-identifier test-instance
```

Sortie :

```
{
    "ValidDBInstanceModificationsMessage": {
        "ValidProcessorFeatures": [],
        "Storage": [
            {
                "StorageSize": [
                    {
                        "Step": 1,
                        "To": 20,
                        "From": 20
                    },
                    {
                        "Step": 1,
                        "To": 6144,
                        "From": 22
                    }
                ],
                "ProvisionedIops": [
                    {
                        "Step": 1,
                        "To": 0,
                        "From": 0
                    }
                ],
                "IopsToStorageRatio": [
                    {
                        "To": 0.0,
                        "From": 0.0
                    }
                ],
                "StorageType": "gp2"
            },
            {
                "StorageSize": [
                    {
                        "Step": 1,
                        "To": 6144,
                        "From": 100
```

```
                    }
                ],
                "ProvisionedIops": [
                    {
                        "Step": 1,
                        "To": 40000,
                        "From": 1000
                    }
                ],
                "IopsToStorageRatio": [
                    {
                        "To": 50.0,
                        "From": 1.0
                    }
                ],
                "StorageType": "io1"
            },
            {
                "StorageSize": [
                    {
                        "Step": 1,
                        "To": 20,
                        "From": 20
                    },
                    {
                        "Step": 1,
                        "To": 3072,
                        "From": 22
                    }
                ],
                "ProvisionedIops": [
                    {
                        "Step": 1,
                        "To": 0,
                        "From": 0
                    }
                ],
                "IopsToStorageRatio": [
                    {
                        "To": 0.0,
                        "From": 0.0
                    }
                ],
                "StorageType": "magnetic"
```

```
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeValidDbInstanceModificationsà la section Référence des AWS CLI commandes.

`download-db-log-file-portion`

L'exemple de code suivant montre comment utiliser`download-db-log-file-portion`.

AWS CLI

Exemple 1 : pour télécharger la dernière partie d'un fichier journal de base de données

L'`download-db-log-file-portion`exemple suivant télécharge uniquement la dernière partie de votre fichier journal et l'enregistre dans un fichier local nommé`tail.txt`.

```
aws rds download-db-log-file-portion \
    --db-instance-identifier test-instance \
    --log-file-name log.txt \
    --output text > tail.txt
```

Le fichier enregistré peut contenir des lignes vides. Ils apparaissent à la fin de chaque partie du fichier journal lors du téléchargement.

Exemple 2 : pour télécharger l'intégralité d'un fichier journal de base de données

L'`download-db-log-file-portion`exemple suivant télécharge l'intégralité du fichier journal à l'aide du `--starting-token 0` paramètre et enregistre la sortie dans un fichier local nommé`full.txt`.

```
aws rds download-db-log-file-portion \
    --db-instance-identifier test-instance \
    --log-file-name log.txt \
    --starting-token 0 \
    --output text > full.txt
```

Le fichier enregistré peut contenir des lignes vides. Ils apparaissent à la fin de chaque partie du fichier journal lors du téléchargement.

- Pour plus de détails sur l'API, reportez-vous DownloadDbLogFilePortionà la section Référence des AWS CLI commandes.

`generate-auth-token`

L'exemple de code suivant montre comment utiliser`generate-auth-token`.

AWS CLI

Pour générer un jeton d'authentification

L'`generate-db-auth-token`exemple suivant génère un jeton d'authentification à utiliser avec l'authentification de base de données IAM.

```
aws rds generate-db-auth-token \
    --hostname aurmysql-test.cdgmuqiadpid.us-west-2.rds.amazonaws.com \
    --port 3306 \
    --region us-east-1 \
    --username jane_doe
```

Sortie :

```
aurmysql-test.cdgmuqiadpid.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIESZCNJ3OEXAMPLE%2F20180731%2Fus-east-1%2Frds-db%2Faws4_request&X-
Amz-Date=20180731T235209Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=5a8753ebEXAMPLEa2c724e5667797EXAMPLE9d6ec6e3f427191fa41aeEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous GenerateAuthTokenà la section Référence des AWS CLI commandes.

`generate-db-auth-token`

L'exemple de code suivant montre comment utiliser`generate-db-auth-token`.

AWS CLI

Pour générer un jeton d'authentification IAM

L'`generate-db-auth-token`exemple suivant génère un jeton d'authentification IAM pour se connecter à une base de données.

```
aws rds generate-db-auth-token \
    --hostname mydb.123456789012.us-east-1.rds.amazonaws.com \
    --port 3306 \
    --region us-east-1 \
    --username db_user
```

Sortie :

```
mydb.123456789012.us-east-1.rds.amazonaws.com:3306/?
Action=connect&DBUser=db_user&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIEXAMPLE%2Fus-east-1%2Frds-db%2Faws4_request&X-Amz-
Date=20210123T011543Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=88987EXAMPLE1EXAMPLE2EXAMPLE3EXAMPLE4EXAMPLE5EXAMPLE6
```

Pour plus d'informations, consultez Connexion à votre instance de base de données à l'aide de l'authentification IAM dans le guide de l'utilisateur Amazon RDS et Connexion à votre cluster de base de données à l'aide de l'authentification IAM dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous GenerateDbAuthTokenà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier des balises sur une ressource Amazon RDS

L'`list-tags-for-resource`exemple suivant répertorie toutes les balises d'une instance de base de données.

```
aws rds list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

Sortie :

```
{
    "TagList": [
```

```
        {
            "Key": "Environment",
            "Value": "test"
        },
        {
            "Key": "Name",
            "Value": "MyDatabase"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources Amazon RDS dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`modify-certificates`

L'exemple de code suivant montre comment utiliser`modify-certificates`.

AWS CLI

Pour remplacer temporairement le SSL/TLS certificat par défaut du système pour les nouvelles instances de base de données

L'`modify-certificates`exemple suivant remplace temporairement le SSL/TLS certificat par défaut du système pour les nouvelles instances de base de données.

```
aws rds modify-certificates \
    --certificate-identifier rds-ca-2019
```

Sortie :

```
{
    "Certificate": {
        "CertificateIdentifier": "rds-ca-2019",
        "CertificateType": "CA",
        "Thumbprint": "EXAMPLE123456789012",
        "ValidFrom": "2019-09-19T18:16:53Z",
        "ValidTill": "2024-08-22T17:08:50Z",
```

```
        "CertificateArn": "arn:aws:rds:us-east-1::cert:rds-ca-2019",
        "CustomerOverride": true,
        "CustomerOverrideValidTill": "2024-08-22T17:08:50Z"
    }
}
```

Pour plus d'informations, consultez Rotation de votre SSL/TLS certificat dans le guide de l'utilisateur Amazon RDS et Rotation de votre SSL/TLS certificat dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyCertificatesà la section Référence des AWS CLI commandes.

`modify-current-db-cluster-capacity`

L'exemple de code suivant montre comment utiliser`modify-current-db-cluster-capacity`.

AWS CLI

Pour augmenter la capacité d'un cluster de base de données Aurora Serverless

L'`modify-current-db-cluster-capacity`exemple suivant augmente la capacité d'un cluster de base de données Aurora Serverless à 8.

```
aws rds modify-current-db-cluster-capacity \
    --db-cluster-identifier mydbcluster \
    --capacity 8
```

Sortie :

```
{
    "DBClusterIdentifier": "mydbcluster",
    "PendingCapacity": 8,
    "CurrentCapacity": 1,
    "SecondsBeforeTimeout": 300,
    "TimeoutAction": "ForceApplyCapacityChange"
}
```

Pour plus d'informations, consultez la section Dimensionnement manuel de la capacité du cluster de base de données Aurora Serverless v1 dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyCurrentDbClusterCapacityà la section Référence des AWS CLI commandes.

`modify-db-cluster-endpoint`

L'exemple de code suivant montre comment utiliser`modify-db-cluster-endpoint`.

AWS CLI

Pour modifier un point de terminaison de cluster de base de données personnalisé

L'`modify-db-cluster-endpoint`exemple suivant modifie le point de terminaison du cluster de base de données personnalisé spécifié.

```
aws rds modify-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint \
    --static-members dbinstance1 dbinstance2 dbinstance3
```

Sortie :

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "modifying",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2",
        "dbinstance3"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

Pour plus d'informations, consultez la section Gestion des connexions Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterEndpointà la section Référence des AWS CLI commandes.

`modify-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`modify-db-cluster-parameter-group`.

AWS CLI

Pour modifier les paramètres d'un groupe de paramètres de cluster de base de données

L'`modify-db-cluster-parameter-group`exemple suivant modifie les valeurs des paramètres d'un groupe de paramètres de cluster de base de données.

```
aws rds modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterpg \
    --
parameters "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate"
 \

  "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Sortie :

```
{
    "DBClusterParameterGroupName": "mydbclusterpg"
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`modify-db-cluster-snapshot-attribute`

L'exemple de code suivant montre comment utiliser`modify-db-cluster-snapshot-attribute`.

AWS CLI

Pour modifier un attribut de capture d'écran d'un cluster de base de données

L'`modify-db-cluster-snapshot-attribute`exemple suivant apporte des modifications à l'attribut de capture d'écran du cluster de base de données spécifié.

```
aws rds modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --attribute-name restore \
    --values-to-add 123456789012
```

Sortie :

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "myclustersnapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012"
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Restauration à partir d'un instantané de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterSnapshotAttributeà la section Référence des AWS CLI commandes.

`modify-db-cluster`

L'exemple de code suivant montre comment utiliser`modify-db-cluster`.

AWS CLI

Exemple 1 : pour modifier un cluster de base de données

L'`modify-db-cluster`exemple suivant modifie le mot de passe de l'utilisateur principal pour le cluster de base de données nommé `cluster-2` et définit la période de conservation des sauvegardes à 14 jours. Le `--apply-immediately` paramètre permet d'effectuer les modifications immédiatement, au lieu d'attendre le créneau de maintenance suivant.

```
aws rds modify-db-cluster \
    --db-cluster-identifier cluster-2 \
    --backup-retention-period 14 \
    --master-user-password newpassword99 \
    --apply-immediately
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "eu-central-1b",
            "eu-central-1c",
            "eu-central-1a"
        ],
        "BackupRetentionPeriod": 14,
        "DatabaseName": "",
        "DBClusterIdentifier": "cluster-2",
        "DBClusterParameterGroup": "default.aurora5.6",
        "DBSubnetGroup": "default-vpc-2305ca49",
        "Status": "available",
        "EarliestRestorableTime": "2020-06-03T02:07:29.637Z",
        "Endpoint": "cluster-2.cluster-############.eu-central-1.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-2.cluster-ro-############.eu-
central-1.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora",
        "EngineVersion": "5.6.10a",
        "LatestRestorableTime": "2020-06-04T15:11:25.748Z",
        "Port": 3306,
        "MasterUsername": "admin",
        "PreferredBackupWindow": "01:55-02:25",
        "PreferredMaintenanceWindow": "thu:21:14-thu:21:44",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [
            {
```

```
                "DBInstanceIdentifier": "cluster-2-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-20a5c047",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1RLNU0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:eu-central-1:123456789012:key/
d1bd7c8f-5cdb-49ca-8a62-a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-AGJ7XI77XVIS6FUXHU1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:eu-central-1:123456789012:cluster:cluster-2",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-04-03T14:44:02.764Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": true,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

Pour plus d'informations, consultez la section Modification d'un cluster de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : pour associer un groupe de sécurité VPC à un cluster de base de données

L'`modify-db-instance`exemple suivant associe un groupe de sécurité VPC spécifique et supprime les groupes de sécurité de base de données d'un cluster de bases de données.

```
aws rds modify-db-cluster \
    --db-cluster-identifier dbName \
    --vpc-security-group-ids sg-ID
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "dbName",
        "DBClusterParameterGroup": "default.aurora-mysql8.0",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2024-02-15T01:12:13.966000+00:00",
        "Endpoint": "dbName.cluster-abcdefghji.us-west-2.rds.amazonaws.com",
        "ReaderEndpoint": "dbName.cluster-ro-abcdefghji.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-mysql",
        "EngineVersion": "8.0.mysql_aurora.3.04.1",
        "LatestRestorableTime": "2024-02-15T02:25:33.696000+00:00",
        "Port": 3306,
        "MasterUsername": "admin",
        "PreferredBackupWindow": "10:59-11:29",
        "PreferredMaintenanceWindow": "thu:08:54-thu:09:24",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "dbName-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-ID",
                "Status": "active"
            }
        ],
        ...output omitted...
    }
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès avec les groupes de sécurité dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbClusterà la section Référence des AWS CLI commandes.

`modify-db-instance`

L'exemple de code suivant montre comment utiliser`modify-db-instance`.

AWS CLI

Exemple 1 : pour modifier une instance de base de données

L'`modify-db-instance`exemple suivant associe un groupe d'options et un groupe de paramètres à une instance de base de données Microsoft SQL Server compatible. Le `--apply-immediately` paramètre entraîne l'association immédiate des groupes d'options et de paramètres, au lieu d'attendre la fenêtre de maintenance suivante.

```
aws rds modify-db-instance \
    --db-instance-identifier database-2 \
    --option-group-name test-se-2017 \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --apply-immediately
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-2",
        "DBInstanceClass": "db.r4.large",
        "Engine": "sqlserver-se",
        "DBInstanceStatus": "available",

        ...output omitted...

        "DBParameterGroups": [
            {
                "DBParameterGroupName": "test-sqlserver-se-2017",
                "ParameterApplyStatus": "applying"
            }
        ],
```

```
        "AvailabilityZone": "us-west-2d",

        ...output omitted...

        "MultiAZ": true,
        "EngineVersion": "14.00.3281.6.v1",
        "AutoMinorVersionUpgrade": false,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "license-included",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "test-se-2017",
                "Status": "pending-apply"
            }
        ],
        "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
        "SecondaryAvailabilityZone": "us-west-2c",
        "PubliclyAccessible": true,
        "StorageType": "gp2",

        ...output omitted...

        "DeletionProtection": false,
        "AssociatedRoles": [],
        "MaxAllocatedStorage": 1000
    }
}
```

Pour plus d'informations, consultez Modifier une instance de base de données Amazon RDS dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour associer un groupe de sécurité VPC à une instance de base de données

L'`modify-db-instance`exemple suivant associe un groupe de sécurité VPC spécifique et supprime les groupes de sécurité de base de données d'une instance de base de données :

```
aws rds modify-db-instance \
    --db-instance-identifier dbName \
    --vpc-security-group-ids sg-ID
```

Sortie :

```
{
```

```
"DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
        "Address": "dbName.abcdefghijk.us-west-2.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "ABCDEFGHIJK1234"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
    "PreferredBackupWindow": "11:57-12:27",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-ID",
            "Status": "active"
        }
    ],
    ... output omitted ...
    "MultiAZ": false,
    "EngineVersion": "8.0.35",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",

    ... output ommited ...
    }
}
```

Pour plus d'informations, consultez la section Contrôle de l'accès avec les groupes de sécurité dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Modifier DBInstance dans AWS CLI la référence des commandes.

`modify-db-parameter-group`

L'exemple de code suivant montre comment utiliser`modify-db-parameter-group`.

AWS CLI

Pour modifier un groupe de paramètres de base de données

L'`modify-db-parameter-group`exemple suivant modifie la valeur du `clr enabled` paramètre dans un groupe de paramètres de base de données. Le `--apply-immediately` paramètre entraîne la modification immédiate du groupe de paramètres de base de données, au lieu d'attendre la fenêtre de maintenance suivante.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Sortie :

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

Pour plus d'informations, consultez la section Modification des paramètres d'un groupe de paramètres de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Modifier DBParameter le groupe dans AWS CLI la référence des commandes.

`modify-db-proxy-endpoint`

L'exemple de code suivant montre comment utiliser`modify-db-proxy-endpoint`.

AWS CLI

Pour modifier un point de terminaison proxy de base de données pour une base de données RDS

L'`modify-db-proxy-endpoint`exemple suivant modifie un point de terminaison du proxy de base de données `proxyEndpoint` pour définir le délai de lecture à 65 secondes.

```
aws rds modify-db-proxy-endpoint \
    --db-proxy-endpoint-name proxyEndpoint \
    --cli-read-timeout 65
```

Sortie :

```
{
"DBProxyEndpoint":
    {
        "DBProxyEndpointName": "proxyEndpoint",
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
        "DBProxyName": "proxyExample",
        "Status": "available",
        "VpcId": "vpc-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyEndpoint.endpoint.proxyExample-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": "false"
    }
}
```

Pour plus d'informations, consultez Modifier un point de terminaison de proxy dans le guide de l'utilisateur Amazon RDS et Modification d'un point de terminaison de proxy dans le guide de l'utilisateur d'Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbProxyEndpointà la section Référence des AWS CLI commandes.

`modify-db-proxy-target-group`

L'exemple de code suivant montre comment utiliser`modify-db-proxy-target-group`.

AWS CLI

Pour modifier les points de terminaison d'un proxy de base de données

L'`modify-db-proxy-target-group`exemple suivant modifie un groupe cible de proxy de base de données pour définir le nombre maximum de connexions à 80 % et le nombre maximum de connexions inactives à 10 %.

```
aws rds modify-db-proxy-target-group \
    --target-group-name default \
    --db-proxy-name proxyExample \
    --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10
```

Sortie :

```
{
"DBProxyTargetGroup":
    {
        "DBProxyName": "proxyExample",
        "TargetGroupName": "default",
        "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
        "IsDefault": true,
        "Status": "available",
        "ConnectionPoolConfig": {
            "MaxConnectionsPercent": 80,
            "MaxIdleConnectionsPercent": 10,
            "ConnectionBorrowTimeout": 120,
            "SessionPinningFilters": []
        },
        "CreatedDate": "2023-05-02T18:41:19.495000+00:00",
        "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

Pour plus d'informations, consultez Modification d'un proxy RDS dans le guide de l'utilisateur Amazon RDS et Modification d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbProxyTargetGroupà la section Référence des AWS CLI commandes.

`modify-db-proxy`

L'exemple de code suivant montre comment utiliser`modify-db-proxy`.

AWS CLI

Pour modifier un proxy de base de données pour une base de données RDS

L'`modify-db-proxy`exemple suivant modifie un proxy de base de données nommé `proxyExample` pour exiger le protocole SSL pour ses connexions.

```
aws rds modify-db-proxy \
    --db-proxy-name proxyExample \
    --require-tls
```

Sortie :

```
{
"DBProxy":
    {
        "DBProxyName": "proxyExample",
        "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
        "Status": "modifying"
        "EngineFamily": "PostgreSQL",
        "VpcId": "sg-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Auth": "[
            {
                "Description": "proxydescription1",
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
            }
        ]",
        "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
        "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
        "RequireTLS": true,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
```

```
}
```

Pour plus d'informations, consultez Modifier un proxy RDS dans le guide de l'utilisateur Amazon RDS et Création d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbProxyà la section Référence des AWS CLI commandes.

`modify-db-shard-group`

L'exemple de code suivant montre comment utiliser`modify-db-shard-group`.

AWS CLI

Exemple 1 : pour modifier un groupe de partitions de base de données

L'`modify-db-shard-group`exemple suivant modifie la capacité maximale d'un groupe de partitions de base de données.

```
aws rds modify-db-shard-group \
    --db-shard-group-identifier my-db-shard-group \
    --max-acu 1000
```

Sortie :

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Aurora DB Clusters dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : pour décrire vos groupes de partitions de base de données

L'`describe-db-shard-groups`exemple suivant récupère les détails de vos groupes de partitions de base de données après avoir exécuté la `modify-db-shard-group` commande. La capacité maximale du groupe de partitions de base de données `my-db-shard-group` est désormais de 1 000 unités de capacité Aurora (ACUs).

```
aws rds describe-db-shard-groups
```

Sortie :

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Aurora DB Clusters dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyDbShardGroupà la section Référence des AWS CLI commandes.

`modify-db-snapshot-attribute`

L'exemple de code suivant montre comment utiliser`modify-db-snapshot-attribute`.

AWS CLI

Exemple 1 : pour permettre à deux AWS comptes de restaurer un instantané de base de données

L'`modify-db-snapshot-attribute`exemple suivant accorde l'autorisation à deux AWS comptes, avec les identifiants `111122223333` et`444455556666`, de restaurer le snapshot de base de données nommé`mydbsnapshot`.

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-add {"111122223333","444455556666"}
```

Sortie :

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333",
                    "444455556666"
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Partage d'un instantané dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour empêcher un AWS compte de restaurer un instantané de base de données

L'`modify-db-snapshot-attribute`exemple suivant supprime l'autorisation d'un AWS compte particulier pour restaurer le snapshot de base de données nommé`mydbsnapshot`. Lorsque vous spécifiez un seul compte, l'identifiant du compte ne peut pas être entouré de guillemets ou d'accolades.

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-remove 444455556666
```

Sortie :

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333"
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Partage d'un instantané dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous ModifyDbSnapshotAttributeà la section Référence des AWS CLI commandes.

`modify-db-snapshot-attributes`

L'exemple de code suivant montre comment utiliser`modify-db-snapshot-attributes`.

AWS CLI

Pour modifier un attribut de capture d'écran de base de données

L'`modify-db-snapshot-attribute`exemple suivant autorise deux identifiants de AWS compte `111122223333` et permet de restaurer `444455556666` le snapshot de base de données nommé`mydbsnapshot`.

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-add '["111122223333","444455556666"]'
```

Sortie :

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333",
                    "444455556666"
                ]
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Partage d'un instantané dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous ModifyDbSnapshotAttributesà la section Référence des AWS CLI commandes.

`modify-db-snapshot`

L'exemple de code suivant montre comment utiliser`modify-db-snapshot`.

AWS CLI

Pour modifier un instantané de base de données

L'`modify-db-snapshot`exemple suivant met à niveau un instantané PostgreSQL 10.6 nommé PostgreSQL `db5-snapshot-upg-test` 11.7. La nouvelle version du moteur de base de

données est affichée une fois que la mise à niveau du snapshot est terminée et que son état est disponible.

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier db5-snapshot-upg-test \
    --engine-version 11.7
```

Sortie :

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "db5-snapshot-upg-test",
        "DBInstanceIdentifier": "database-5",
        "SnapshotCreateTime": "2020-03-27T20:49:17.092Z",
        "Engine": "postgres",
        "AllocatedStorage": 20,
        "Status": "upgrading",
        "Port": 5432,
        "AvailabilityZone": "us-west-2a",
        "VpcId": "vpc-2ff27557",
        "InstanceCreateTime": "2020-03-27T19:59:04.735Z",
        "MasterUsername": "postgres",
        "EngineVersion": "10.6",
        "LicenseModel": "postgresql-license",
        "SnapshotType": "manual",
        "OptionGroupName": "default:postgres-11",
        "PercentProgress": 100,
        "StorageType": "gp2",
        "Encrypted": false,
        "DBSnapshotArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
upg-test",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-GJMF75LM42IL6BTFRE4UZJ5YM4"
    }
}
```

Pour plus d'informations, consultez la section Mise à niveau d'un instantané de base de données PostgreSQL dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous ModifyDbSnapshotà la section Référence des AWS CLI commandes.

`modify-db-subnet-group`

L'exemple de code suivant montre comment utiliser`modify-db-subnet-group`.

AWS CLI

Pour modifier un groupe de sous-réseaux de base de données

L'`modify-db-subnet-group`exemple suivant ajoute un sous-réseau avec cet ID `subnet-08e41f9e230222222` au groupe de sous-réseaux de base de données nommé. `mysubnetgroup` Pour conserver les sous-réseaux existants dans le groupe de sous-réseaux, incluez leurs IDs valeurs dans l'`--subnet-ids`option. Assurez-vous que le groupe de sous-réseaux de base de données comporte au moins deux zones de disponibilité différentes.

```
aws rds modify-db-subnet-group \
    --db-subnet-group-name mysubnetgroup \
    --subnet-ids
 '["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef","subnet-
```

Sortie :

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mysubnetgroup",
        "DBSubnetGroupDescription": "test DB subnet group",
        "VpcId": "vpc-0f08e7610a1b2c3d4",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-08e41f9e230222222",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            },
            {
```

```
                "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ],
        "DBSubnetGroupArn": "arn:aws:rds:us-
west-2:534026745191:subgrp:mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez Étape 3 : Création d'un groupe de sous-réseaux de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous ModifyDbSubnetGroupà la section Référence des AWS CLI commandes.

`modify-event-subscription`

L'exemple de code suivant montre comment utiliser`modify-event-subscription`.

AWS CLI

Pour modifier l'abonnement à un événement

L'`modify-event-subscription`exemple suivant désactive l'abonnement à un événement spécifié, de sorte qu'il ne publie plus de notifications sur la rubrique Amazon Simple Notification Service spécifiée.

```
aws rds modify-event-subscription \
    --subscription-name my-instance-events \
    --no-enabled
```

Sortie :

```
{
    "EventSubscription": {
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "CustomerAwsId": "123456789012",
        "SourceType": "db-instance",
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "CustSubscriptionId": "my-instance-events",
        "Status": "modifying",
        "Enabled": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyEventSubscriptionà la section Référence des AWS CLI commandes.

`modify-global-cluster`

L'exemple de code suivant montre comment utiliser`modify-global-cluster`.

AWS CLI

Pour modifier un cluster de bases de données global

L'`modify-global-cluster`exemple suivant active la protection contre les suppressions pour un cluster de base de données global compatible Aurora MySQL.

```
aws rds modify-global-cluster \
    --global-cluster-identifier myglobalcluster \
    --deletion-protection
```

Sortie :

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
```

```
        "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.mysql_aurora.2.07.2",
        "StorageEncrypted": false,
        "DeletionProtection": true,
        "GlobalClusterMembers": []
    }
}
```

Pour plus d'informations, consultez la section Gestion d'une base de données globale Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ModifyGlobalClusterà la section Référence des AWS CLI commandes.

`promote-read-replica-db-cluster`

L'exemple de code suivant montre comment utiliser`promote-read-replica-db-cluster`.

AWS CLI

Pour promouvoir un cluster de bases de données, lisez la réplique

L'`promote-read-replica-db-cluster`exemple suivant fait de la réplique de lecture spécifiée un cluster de base de données autonome.

```
aws rds promote-read-replica-db-cluster \
    --db-cluster-identifier mydbcluster-1
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
```

```
        "BackupRetentionPeriod": 1,
        "DatabaseName": "",
        "DBClusterIdentifier": "mydbcluster-1",
        ...some output truncated...
    }
}
```

Pour plus d'informations, consultez la section Promouvoir une réplique en lecture en tant que cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous PromoteReadReplicaDbClusterà la section Référence des AWS CLI commandes.

`promote-read-replica`

L'exemple de code suivant montre comment utiliser`promote-read-replica`.

AWS CLI

Pour promouvoir une réplique lue

L'`promote-read-replica`exemple suivant fait de la réplique de lecture spécifiée une instance de base de données autonome.

```
aws rds promote-read-replica \
    --db-instance-identifier test-instance-repl
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
        "StorageType": "standard",
        "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "modifying",
        ...some output truncated...
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PromoteReadReplicaà la section Référence des AWS CLI commandes.

`purchase-reserved-db-instance`

L'exemple de code suivant montre comment utiliser`purchase-reserved-db-instance`.

AWS CLI

Pour acheter une offre d'instance de base de données réservée

L'`purchase-reserved-db-instances-offering`exemple suivant achète une offre d'instance de base de données réservée. `reserved-db-instances-offering-id`Il doit s'agir d'un identifiant d'offre valide, tel que renvoyé par la `describe-reserved-db-instances-offering` commande.

aws rds purchase-reserved-db-instances -offering -- reserved-db-instances-offering -id 438012d3-4a52-4cc7-b2e3-8dff72e0e706

- Pour plus de détails sur l'API, reportez-vous PurchaseReservedDbInstanceà la section Référence des AWS CLI commandes.

`purchase-reserved-db-instances-offerings`

L'exemple de code suivant montre comment utiliser`purchase-reserved-db-instances-offerings`.

AWS CLI

Exemple 1 : pour trouver une instance de base de données réservée à acheter

L'`describe-reserved-db-instances-offerings`exemple suivant répertorie les instances de base de données MySQL réservées disponibles avec la classe d'instance db.t2.micro et pour une durée d'un an. L'ID d'offre est requis pour acheter une instance de base de données réservée.

```
aws rds describe-reserved-db-instances-offerings \
    --product-description mysql \
    --db-instance-class db.t2.micro \
    --duration 1
```

Sortie :

```
{
    "ReservedDBInstancesOfferings": [
```

```
        {
            "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
            "DBInstanceClass": "db.t2.micro",
            "Duration": 31536000,
            "FixedPrice": 51.0,
            "UsagePrice": 0.0,
            "CurrencyCode": "USD",
            "ProductDescription": "mysql",
            "OfferingType": "Partial Upfront",
            "MultiAZ": false,
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.006,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        },
    ... some output truncated ...
    ]
}
```

Pour plus d'informations, consultez la section Instances de base de données réservées pour Amazon RDS dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour acheter une instance de base de données réservée

L'`purchase-reserved-db-instances-offering`exemple suivant montre comment acheter l'offre d'instance de base de données réservée de l'exemple précédent.

aws rds purchase-reserved-db-instances -offering -- -id 8ba30be1-b9ec-447f-8f23-6114e3f4c7b4 reserved-db-instances-offering

Sortie :

```
{
    "ReservedDBInstance": {
        "ReservedDBInstanceId": "ri-2020-06-29-16-54-57-670",
        "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
        "DBInstanceClass": "db.t2.micro",
        "StartTime": "2020-06-29T16:54:57.670Z",
        "Duration": 31536000,
        "FixedPrice": 51.0,
        "UsagePrice": 0.0,
```

```
        "CurrencyCode": "USD",
        "DBInstanceCount": 1,
        "ProductDescription": "mysql",
        "OfferingType": "Partial Upfront",
        "MultiAZ": false,
        "State": "payment-pending",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.006,
                "RecurringChargeFrequency": "Hourly"
            }
        ],
        "ReservedDBInstanceArn": "arn:aws:rds:us-
west-2:123456789012:ri:ri-2020-06-29-16-54-57-670"
    }
}
```

Pour plus d'informations, consultez la section Instances de base de données réservées pour Amazon RDS dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous PurchaseReservedDbInstancesOfferingsà la section Référence des AWS CLI commandes.

`reboot-db-instance`

L'exemple de code suivant montre comment utiliser`reboot-db-instance`.

AWS CLI

Pour redémarrer une instance de base de données

L'`reboot-db-instance`exemple suivant lance un redémarrage de l'instance de base de données spécifiée.

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
```

```
        "Engine": "mysql",
        "DBInstanceStatus": "rebooting",
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.############.us-
west-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z1PVIF0EXAMPLE"
        },

    ... output omitted...

    }
}
```

Pour plus d'informations, consultez la section Redémarrage d'une instance de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, voir Reboot DBInstance dans AWS CLI la référence des commandes.

`reboot-db-shard-group`

L'exemple de code suivant montre comment utiliser`reboot-db-shard-group`.

AWS CLI

Exemple 1 : pour redémarrer un groupe de partitions de base de données

L'`reboot-db-shard-group`exemple suivant redémarre un groupe de partitions de base de données.

```
aws rds reboot-db-shard-group \
    --db-shard-group-identifier my-db-shard-group
```

Sortie :

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
```

```
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, consultez la section Redémarrage d'un cluster de base de données Amazon Aurora ou d'une instance de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : pour décrire vos groupes de partitions de base de données

L'`describe-db-shard-groups`exemple suivant récupère les détails de vos groupes de partitions de base de données après avoir exécuté la `reboot-db-shard-group` commande. Le groupe de partitions de base de données `my-db-shard-group` est en train de redémarrer.

```
aws rds describe-db-shard-groups
```

Sortie :

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 1000.0,
```

```
            "ComputeRedundancy": 0,
            "Status": "rebooting",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

Pour plus d'informations, consultez la section Redémarrage d'un cluster de base de données Amazon Aurora ou d'une instance de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RebootDbShardGroupà la section Référence des AWS CLI commandes.

`register-db-proxy-targets`

L'exemple de code suivant montre comment utiliser`register-db-proxy-targets`.

AWS CLI

Pour enregistrer un proxy de base de données auprès d'une base de données

L'`register-db-proxy-targets`exemple suivant crée l'association entre une base de données et un proxy.

```
aws rds register-db-proxy-targets \
    --db-proxy-name proxyExample \
    --db-cluster-identifiers database-5
```

Sortie :

```
{
    "DBProxyTargets": [
        {
            "RdsResourceId": "database-5",
            "Port": 3306,
            "Type": "TRACKED_CLUSTER",
            "TargetHealth": {
                "State": "REGISTERING"
            }
```

```
        },
        {
            "Endpoint": "database-5instance-1.ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RdsResourceId": "database-5",
            "Port": 3306,
            "Type": "RDS_INSTANCE",
            "TargetHealth": {
                "State": "REGISTERING"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Création d'un proxy RDS dans le guide de l'utilisateur Amazon RDS et Création d'un proxy RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RegisterDbProxyTargetsà la section Référence des AWS CLI commandes.

`remove-from-global-cluster`

L'exemple de code suivant montre comment utiliser`remove-from-global-cluster`.

AWS CLI

Pour détacher un cluster secondaire Aurora d'un cluster de base de données global Aurora

L'`remove-from-global-cluster`exemple suivant détache un cluster secondaire Aurora d'un cluster de base de données global Aurora. Le cluster passe d'un cluster en lecture seule à un cluster autonome doté d'une capacité de lecture-écriture.

```
aws rds remove-from-global-cluster \
    --region us-west-2 \
    --global-cluster-identifier myglobalcluster \
    --db-cluster-identifier arn:aws:rds:us-west-2:123456789012:cluster:DB-1
```

Sortie :

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
```

```
        "GlobalClusterResourceId": "cluster-abc123def456gh",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.11",
        "StorageEncrypted": true,
        "DeletionProtection": false,
        "GlobalClusterMembers": [
            {
                "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:js-
global-cluster",
                "Readers": [
                    "arn:aws:rds:us-west-2:123456789012:cluster:DB-1"
                ],
                "IsWriter": true
            },
            {
                "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:DB-1",
                "Readers": [],
                "IsWriter": false,
                "GlobalWriteForwardingStatus": "disabled"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Suppression d'un cluster d'une base de données globale Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RemoveFromGlobalClusterà la section Référence des AWS CLI commandes.

`remove-option-from-option-group`

L'exemple de code suivant montre comment utiliser`remove-option-from-option-group`.

AWS CLI

Pour supprimer une option d'un groupe d'options

L'`remove-option-from-option-group`exemple suivant supprime l'`OEM`option de`myoptiongroup`.

```
aws rds remove-option-from-option-group \
    --option-group-name myoptiongroup \
    --options OEM \
    --apply-immediately
```

Sortie :

```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Test",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "19",
        "Options": [],
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
    }
}
```

Pour plus d'informations, consultez Supprimer une option d'un groupe d'options dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RemoveOptionFromOptionGroupà la section Référence des AWS CLI commandes.

`remove-role-from-db-cluster`

L'exemple de code suivant montre comment utiliser`remove-role-from-db-cluster`.

AWS CLI

Pour dissocier un rôle AWS Identity and Access Management (IAM) d'un cluster de bases de données

L'`remove-role-from-db-cluster`exemple suivant supprime un rôle d'un cluster de base de données.

```
aws rds remove-role-from-db-cluster \
    --db-cluster-identifier mydbcluster \
    --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Associer un rôle IAM à un cluster de bases de données Amazon Aurora MySQL dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RemoveRoleFromDbClusterà la section Référence des AWS CLI commandes.

`remove-role-from-db-instance`

L'exemple de code suivant montre comment utiliser`remove-role-from-db-instance`.

AWS CLI

Pour dissocier un rôle AWS Identity and Access Management (IAM) d'une instance de base de données

L'`remove-role-from-db-instance`exemple suivant supprime le rôle nommé d'une instance `rds-s3-integration-role` de base de données Oracle nomméé`test-instance`.

```
aws rds remove-role-from-db-instance \
    --db-instance-identifier test-instance \
    --feature-name S3_INTEGRATION \
    --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation de l'intégration de RDS SQL Server à S3 dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous RemoveRoleFromDbInstanceà la section Référence des AWS CLI commandes.

`remove-source-identifier-from-subscription`

L'exemple de code suivant montre comment utiliser`remove-source-identifier-from-subscription`.

AWS CLI

Pour supprimer un identifiant de source d'un abonnement

L'`remove-source-identifier`exemple suivant supprime l'identifiant de source spécifié d'un abonnement existant.

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name my-instance-events \
    --source-identifier test-instance-repl
```

Sortie :

```
{
    "EventSubscription": {
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "Status": "modifying",
        "CustSubscriptionId": "my-instance-events",
        "CustomerAwsId": "123456789012",
        "SourceIdsList": [
            "test-instance"
        ],
        "SourceType": "db-instance",
        "Enabled": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous RemoveSourceIdentifierFromSubscriptionà la section Référence des AWS CLI commandes.

`remove-tags-from-resource`

L'exemple de code suivant montre comment utiliser`remove-tags-from-resource`.

AWS CLI

Pour supprimer des balises d'une ressource

L'`remove-tags-from-resource`exemple suivant supprime les balises d'une ressource.

```
aws rds remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \
    --tag-keys Name Environment
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Balisage des ressources Amazon RDS dans le guide de l'utilisateur Amazon RDS et balisage des ressources Amazon RDS dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RemoveTagsFromResourceà la section Référence des AWS CLI commandes.

`reset-db-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`reset-db-cluster-parameter-group`.

AWS CLI

Exemple 1 : pour rétablir les valeurs par défaut de tous les paramètres

L'`reset-db-cluster-parameter-group`exemple suivant rétablit les valeurs par défaut de toutes les valeurs de paramètres d'un groupe de paramètres de cluster de base de données créé par le client.

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclpg \
    --reset-all-parameters
```

Sortie :

```
{
    "DBClusterParameterGroupName": "mydbclpg"
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : pour rétablir les valeurs par défaut de certains paramètres

L'`reset-db-cluster-parameter-group`exemple suivant rétablit les valeurs par défaut des paramètres spécifiques dans un groupe de paramètres de cluster de base de données créé par le client.

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclpgy \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Sortie :

```
{
    "DBClusterParameterGroupName": "mydbclpg"
}
```

Pour plus d'informations, consultez la section Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de bases de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ResetDbClusterParameterGroupà la section Référence des AWS CLI commandes.

`reset-db-parameter-group`

L'exemple de code suivant montre comment utiliser`reset-db-parameter-group`.

AWS CLI

Exemple 1 : pour rétablir les valeurs par défaut de tous les paramètres

L'`reset-db-parameter-group`exemple suivant rétablit les valeurs par défaut de toutes les valeurs de paramètres d'un groupe de paramètres de base de données créé par le client.

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mypg \
    --reset-all-parameters
```

Sortie :

```
{
```

```
    "DBParameterGroupName": "mypg"
}
```

Pour plus d'informations, consultez les sections Utilisation des groupes de paramètres de base de données dans le guide de l'utilisateur Amazon RDS et Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

Exemple 2 : pour rétablir les valeurs par défaut de certains paramètres

L'`reset-db-parameter-group`exemple suivant rétablit les valeurs par défaut des paramètres spécifiques dans un groupe de paramètres de base de données créé par le client.

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mypg \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Sortie :

```
{
    "DBParameterGroupName": "mypg"
}
```

Pour plus d'informations, consultez les sections Utilisation des groupes de paramètres de base de données dans le guide de l'utilisateur Amazon RDS et Utilisation des groupes de paramètres de base de données et des groupes de paramètres de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous ResetDbParameterGroupà la section Référence des AWS CLI commandes.

`restore-db-cluster-from-s3`

L'exemple de code suivant montre comment utiliser`restore-db-cluster-from-s3`.

AWS CLI

Pour restaurer un cluster de base de données Amazon Aurora à partir d'Amazon S3

L'`restore-db-cluster-from-s3`exemple suivant restaure un cluster de base de données compatible avec Amazon Aurora MySQL version 5.7 à partir d'un fichier de sauvegarde de base de données MySQL 5.7 dans Amazon S3.

```
aws rds restore-db-cluster-from-s3 \
    --db-cluster-identifier cluster-s3-restore \
    --engine aurora-mysql \
    --master-username admin \
    --master-user-password mypassword \
    --s3-bucket-name amzn-s3-demo-bucket \
    --s3-prefix test-backup \
    --s3-ingestion-role-arn arn:aws:iam::123456789012:role/service-role/TestBackup \
    --source-engine mysql \
    --source-engine-version 5.7.28
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "cluster-s3-restore",
        "DBClusterParameterGroup": "default.aurora-mysql5.7",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "cluster-s3-restore.cluster-co3xyzabc123.us-
west-2.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-s3-restore.cluster-ro-co3xyzabc123.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.12",
        "Port": 3306,
        "MasterUsername": "admin",
        "PreferredBackupWindow": "11:15-11:45",
        "PreferredMaintenanceWindow": "thu:12:19-thu:12:49",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
```

```
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": false,
        "DbClusterResourceId": "cluster-SU5THYQQHOWCXZZDGXREXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:cluster-s3-
restore",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-07-27T14:22:08.095Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

Pour plus d'informations, consultez la section Migration de données depuis MySQL à l'aide d'un compartiment Amazon S3 dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, consultez RestoreDbClusterFromS3 dans AWS CLI Command Reference.

`restore-db-cluster-from-snapshot`

L'exemple de code suivant montre comment utiliser`restore-db-cluster-from-snapshot`.

AWS CLI

Pour restaurer un cluster de bases de données à partir d'un instantané

Ce qui suit `restore-db-cluster-from-snapshot` restaure un cluster de base de données Aurora PostgreSQL compatible avec PostgreSQL version 10.7 à partir d'un instantané de cluster de base de données nommé. `test-instance-snapshot`

```
aws rds restore-db-cluster-from-snapshot \
```

```
    --db-cluster-identifier newdbcluster \
    --snapshot-identifier test-instance-snapshot \
    --engine aurora-postgresql \
    --engine-version 10.7
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "newdbcluster",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "newdbcluster.cluster-############.us-west-2.rds.amazonaws.com",
        "ReaderEndpoint": "newdbcluster.cluster-ro-############.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.7",
        "Port": 5432,
        "MasterUsername": "postgres",
        "PreferredBackupWindow": "09:33-10:03",
        "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
```

```
        "DbClusterResourceId": "cluster-5DSB5IFQDDUVAWOUWM1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:newdbcluster",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-06-05T15:06:58.634Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

Pour plus d'informations, consultez la section Restauration à partir d'un instantané de cluster de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RestoreDbClusterFromSnapshotà la section Référence des AWS CLI commandes.

`restore-db-cluster-to-point-in-time`

L'exemple de code suivant montre comment utiliser`restore-db-cluster-to-point-in-time`.

AWS CLI

Pour restaurer un cluster de bases de données à une heure spécifiée

L'`restore-db-cluster-to-point-in-time`exemple suivant restaure le cluster de base de données nommé `database-4` à l'heure la plus récente possible. L'utilisation `copy-on-write` comme type de restauration restaure le nouveau cluster de base de données en tant que clone du cluster de base de données source.

```
aws rds restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier database-4 \
    --db-cluster-identifier sample-cluster-clone \
    --restore-type copy-on-write \
    --use-latest-restorable-time
```

Sortie :

```
{
```

```
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "sample-cluster-clone",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-clone.cluster-############.us-
west-2.rds.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-clone.cluster-ro-############.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.7",
        "Port": 5432,
        "MasterUsername": "postgres",
        "PreferredBackupWindow": "09:33-10:03",
        "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-BIZ77GDSA2XBSTNPFW1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
clone",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "CloneGroupId": "8d19331a-099a-45a4-b4aa-11aa22bb33cc44dd",
        "ClusterCreateTime": "2020-03-10T19:57:38.967Z",
        "EngineMode": "provisioned",
```

```
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false
    }
}
```

Pour plus d'informations, consultez la section Restauration d'un cluster de base de données à une heure spécifiée dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous RestoreDbClusterToPointInTimeà la section Référence des AWS CLI commandes.

`restore-db-instance-from-db-snapshot`

L'exemple de code suivant montre comment utiliser`restore-db-instance-from-db-snapshot`.

AWS CLI

Pour restaurer une instance de base de données à partir d'un instantané de base de données

L'`restore-db-instance-from-db-snapshot`exemple suivant crée une nouvelle instance de base de données nommée `db7-new-instance` avec la classe d'`db.t3.small`instance de base de données à partir de l'instantané de base de données spécifié. L'instance de base de données source à partir de laquelle le cliché a été pris utilise une classe d'instance de base de données obsolète, vous ne pouvez donc pas la mettre à niveau.

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier db7-new-instance \
    --db-snapshot-identifier db7-test-snapshot \
    --db-instance-class db.t3.small
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "db7-new-instance",
        "DBInstanceClass": "db.t3.small",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",

        ...output omitted...
```

```
        "PreferredMaintenanceWindow": "mon:07:37-mon:08:07",
        "PendingModifiedValues": {},
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",

        ...output omitted...

        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:db7-new-instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

Pour plus d'informations, consultez la section Restauration à partir d'un instantané de base de données dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous RestoreDbInstanceFromDbSnapshotà la section Référence des AWS CLI commandes.

`restore-db-instance-from-s3`

L'exemple de code suivant montre comment utiliser`restore-db-instance-from-s3`.

AWS CLI

Pour restaurer une instance de base de données à partir d'une sauvegarde dans Amazon S3

L'`restore-db-instance-from-s3`exemple suivant crée une nouvelle instance de base de données nommée `restored-test-instance` à partir d'une sauvegarde existante dans le compartiment `my-backups` S3.

```
aws rds restore-db-instance-from-s3 \
    --db-instance-identifier restored-test-instance \
    --allocated-storage 250 --db-instance-class db.m4.large --engine mysql \
    --master-username master --master-user-password secret99 \
    --s3-bucket-name my-backups --s3-ingestion-role-
arn arn:aws:iam::123456789012:role/my-role \
```

```
    --source-engine mysql --source-engine-version 5.6.27
```

- Pour plus de détails sur l'API, consultez RestoreDbInstanceFromS3 dans AWS CLI Command Reference.

`restore-db-instance-to-point-in-time`

L'exemple de code suivant montre comment utiliser`restore-db-instance-to-point-in-time`.

AWS CLI

Exemple 1 : pour restaurer une instance de base de données à un moment donné

L'`restore-db-instance-to-point-in-time`exemple suivant restaure `test-instance` une nouvelle instance de base de données nommée`restored-test-instance`, à l'heure spécifiée.

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier test-instance \
    --target-db-instance restored-test-instance \
    --restore-time 2018-07-30T23:45:00.000Z
```

Sortie :

```
{
    "DBInstance": {
        "AllocatedStorage": 20,
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-
instance",
        "DBInstanceStatus": "creating",
        "DBInstanceIdentifier": "restored-test-instance",
        ...some output omitted...
    }
}
```

Pour plus d'informations, consultez Restaurer une instance de base de données à une heure spécifiée dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : pour restaurer une instance de base de données à une heure spécifiée à partir d'une sauvegarde répliquée

L'`restore-db-instance-to-point-in-time`exemple suivant restaure une instance de base de données Oracle à l'heure spécifiée à partir d'une sauvegarde automatique répliquée.

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example" \
    --target-db-instance-identifier myorclinstance-from-replicated-backup \
    --restore-time 2020-12-08T18:45:00.000Z
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "myorclinstance-from-replicated-backup",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "oracle-se2",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "DBName": "ORCL",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "07:45-08:15",
        "BackupRetentionPeriod": 14,
        ... some output omitted ...
        "DbiResourceId": "db-KGLXG75BGVIWKQT7NQ4EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:myorclinstance-from-
replicated-backup",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": [],
        "TagList": []
    }
}
```

Pour plus d'informations, consultez la section Restauration à une heure spécifiée à partir d'une sauvegarde répliquée dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous RestoreDbInstanceToPointInTimeà la section Référence des AWS CLI commandes.

`start-activity-stream`

L'exemple de code suivant montre comment utiliser`start-activity-stream`.

AWS CLI

Pour démarrer un flux d'activité de base de données

L'`start-activity-stream`exemple suivant démarre un flux d'activité asynchrone pour surveiller un cluster Aurora nommé. my-pg-cluster

```
aws rds start-activity-stream \
    --region us-east-1 \
    --mode async \
    --kms-key-id arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1 \
    --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \
    --apply-immediately
```

Sortie :

```
{
    "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
    "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGHI1JKLM2NOPQ3R4S",
    "Status": "starting",
    "Mode": "async",
    "ApplyImmediately": true
}
```

Pour plus d'informations, consultez la section Démarrage d'un flux d'activité de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous StartActivityStreamà la section Référence des AWS CLI commandes.

`start-db-cluster`

L'exemple de code suivant montre comment utiliser`start-db-cluster`.

AWS CLI

Pour démarrer un cluster de bases de données

L'`start-db-cluster`exemple suivant démarre un cluster de base de données et ses instances de base de données.

```
aws rds start-db-cluster \
    --db-cluster-identifier mydbcluster
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1e",
            "us-east-1b"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "mydb",
        "DBClusterIdentifier": "mydbcluster",
        ...some output truncated...
    }
}
```

Pour plus d'informations, consultez la section Arrêt et démarrage d'un cluster de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous StartDbClusterà la section Référence des AWS CLI commandes.

`start-db-instance-automated-backups-replication`

L'exemple de code suivant montre comment utiliser`start-db-instance-automated-backups-replication`.

AWS CLI

Pour activer les sauvegardes automatisées entre régions

L'`start-db-instance-automated-backups-replication`exemple suivant réplique des sauvegardes automatisées d'une instance de base de données située dans la région USA Est (Virginie du Nord) vers USA Ouest (Oregon). La période de conservation des sauvegardes est de 14 jours.

```
aws rds start-db-instance-automated-backups-replication \
    --region us-west-2 \
    --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db" \
    --backup-retention-period 14
```

Sortie :

```
{
    "DBInstanceAutomatedBackup": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
        "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
        "Region": "us-east-1",
        "DBInstanceIdentifier": "new-orcl-db",
        "RestoreWindow": {},
        "AllocatedStorage": 20,
        "Status": "pending",
        "Port": 1521,
        "InstanceCreateTime": "2020-12-04T15:28:31Z",
        "MasterUsername": "admin",
        "Engine": "oracle-se2",
        "EngineVersion": "12.1.0.2.v21",
        "LicenseModel": "bring-your-own-license",
        "OptionGroupName": "default:oracle-se2-12-1",
        "Encrypted": false,
        "StorageType": "gp2",
        "IAMDatabaseAuthenticationEnabled": false,
        "BackupRetentionPeriod": 14,
        "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

Pour plus d'informations, consultez la section Activation des sauvegardes automatisées entre régions dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous StartDbInstanceAutomatedBackupsReplicationà la section Référence des AWS CLI commandes.

`start-db-instance`

L'exemple de code suivant montre comment utiliser`start-db-instance`.

AWS CLI

Pour démarrer une instance de base de données

L'`start-db-instance`exemple suivant démarre l'instance de base de données spécifiée.

```
aws rds start-db-instance \
    --db-instance-identifier test-instance
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceStatus": "starting",
        ...some output truncated...
    }
}
```

- Pour plus de détails sur l'API, reportez-vous StartDbInstanceà la section Référence des AWS CLI commandes.

`start-export-task`

L'exemple de code suivant montre comment utiliser`start-export-task`.

AWS CLI

Pour exporter un instantané vers Amazon S3

L'`start-export-task`exemple suivant exporte un instantané de base `db5-snapshot-test` de données nommé dans le compartiment Amazon S3 nomméamzn-s3-demo-bucket.

```
aws rds start-export-task \
    --export-task-identifier my-s3-export \
    --source-arn arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test \
    --s3-bucket-name amzn-s3-demo-bucket \
    --iam-role-arn arn:aws:iam::123456789012:role/service-role/ExportRole \
    --kms-key-id arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-
aabbccddeeff
```

Sortie :

```
{
    "ExportTaskIdentifier": "my-s3-export",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test",
    "SnapshotTime": "2020-03-27T20:48:42.023Z",
    "S3Bucket": "amzn-s3-demo-bucket",
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-
aabbccddeeff",
    "Status": "STARTING",
    "PercentProgress": 0,
    "TotalExtractedDataInGB": 0
}
```

Pour plus d'informations, consultez la section Exportation d'un instantané vers un compartiment Amazon S3 dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous StartExportTaskà la section Référence des AWS CLI commandes.

`stop-activity-stream`

L'exemple de code suivant montre comment utiliser`stop-activity-stream`.

AWS CLI

Pour arrêter le flux d'activité d'une base de données

L'`stop-activity-stream`exemple suivant arrête un flux d'activité dans un cluster Aurora nommé my-pg-cluster.

```
aws rds stop-activity-stream \
    --region us-east-1 \
    --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \
    --apply-immediately
```

Sortie :

```
{
    "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
    "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGHI1JKLM2NOPQ3R4S",
    "Status": "stopping"
```

```
}
```

Pour plus d'informations, consultez la section Arrêter un flux d'activité dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous StopActivityStreamà la section Référence des AWS CLI commandes.

`stop-db-cluster`

L'exemple de code suivant montre comment utiliser`stop-db-cluster`.

AWS CLI

Pour arrêter un cluster de bases de données

L'`stop-db-cluster`exemple suivant arrête un cluster de base de données et ses instances de base de données.

```
aws rds stop-db-cluster \
    --db-cluster-identifier mydbcluster
```

Sortie :

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1e",
            "us-east-1b"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "mydb",
        "DBClusterIdentifier": "mydbcluster",
        ...some output truncated...
    }
}
```

Pour plus d'informations, consultez la section Arrêt et démarrage d'un cluster de base de données Amazon Aurora dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous StopDbClusterà la section Référence des AWS CLI commandes.

`stop-db-instance-automated-backups-replication`

L'exemple de code suivant montre comment utiliser`stop-db-instance-automated-backups-replication`.

AWS CLI

Pour arrêter de répliquer les sauvegardes automatisées

Ce qui suit `stop-db-instance-automated-backups-replication` met fin à la réplication des sauvegardes automatisées dans la région de l'ouest des États-Unis (Oregon). Les sauvegardes répliquées sont conservées conformément à la période de conservation des sauvegardes définie.

```
aws rds stop-db-instance-automated-backups-replication \
    --region us-west-2 \
    --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db"
```

Sortie :

```
{
    "DBInstanceAutomatedBackup": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
        "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
        "Region": "us-east-1",
        "DBInstanceIdentifier": "new-orcl-db",
        "RestoreWindow": {
            "EarliestTime": "2020-12-04T23:13:21.030Z",
            "LatestTime": "2020-12-07T19:59:57Z"
        },
        "AllocatedStorage": 20,
        "Status": "replicating",
        "Port": 1521,
        "InstanceCreateTime": "2020-12-04T15:28:31Z",
        "MasterUsername": "admin",
        "Engine": "oracle-se2",
        "EngineVersion": "12.1.0.2.v21",
        "LicenseModel": "bring-your-own-license",
```

```
        "OptionGroupName": "default:oracle-se2-12-1",
        "Encrypted": false,
        "StorageType": "gp2",
        "IAMDatabaseAuthenticationEnabled": false,
        "BackupRetentionPeriod": 7,
        "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

Pour plus d'informations, consultez la section Arrêter la réplication automatique des sauvegardes dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous StopDbInstanceAutomatedBackupsReplicationà la section Référence des AWS CLI commandes.

`stop-db-instance`

L'exemple de code suivant montre comment utiliser`stop-db-instance`.

AWS CLI

Pour arrêter une instance de base de données

L'`stop-db-instance`exemple suivant arrête l'instance de base de données spécifiée.

```
aws rds stop-db-instance \
    --db-instance-identifier test-instance
```

Sortie :

```
{
    "DBInstance": {
        "DBInstanceStatus": "stopping",
        ...some output truncated...
    }
}
```

- Pour plus de détails sur l'API, reportez-vous StopDbInstanceà la section Référence des AWS CLI commandes.

`switchover-blue-green-deployment`

L'exemple de code suivant montre comment utiliser`switchover-blue-green-deployment`.

AWS CLI

Exemple 1 : pour changer de blue/green déploiement pour une instance de base de données RDS

L'`switchover-blue-green-deployment`exemple suivant fait la promotion de l'environnement vert spécifié en tant que nouvel environnement de production.

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --switchover-timeout 300
```

Sortie :

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
        "BlueGreenDeploymentName": "bgd-cli-test-instance",
        "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
blhi1e",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-blhi1e",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-k5fv7u",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
```

```
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ggsh8m",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-o2vwm0",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CONFIGURE_BACKUPS",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                "Status": "COMPLETED"
            }
        ],
        "Status": "SWITCHOVER_IN_PROGRESS",
        "CreateTime": "2022-02-25T22:33:22.225000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Changer de blue/green déploiement dans le guide de l'utilisateur Amazon RDS.

Exemple 2 : Pour promouvoir un blue/green déploiement pour un cluster de bases de données Aurora MySQL

L'`switchover-blue-green-deployment`exemple suivant fait la promotion de l'environnement vert spécifié en tant que nouvel environnement de production.

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --switchover-timeout 300
```

Sortie :

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3ud8z6",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3ud8z6",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-bvxc73",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-7wc4ie",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-p4xxkz",
                "Status": "AVAILABLE"
            },
```

```
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-np1ikl",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-miszlf",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_CUSTOM_ENDPOINTS",
                "Status": "COMPLETED"
            }
        ],
        "Status": "SWITCHOVER_IN_PROGRESS",
        "CreateTime": "2022-02-25T22:38:49.522000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Changer de blue/green déploiement dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, reportez-vous SwitchoverBlueGreenDeploymentà la section Référence des AWS CLI commandes.

Exemples d'Amazon RDS Data Service utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon RDS Data Service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-execute-statement`

L'exemple de code suivant montre comment utiliser`batch-execute-statement`.

AWS CLI

Pour exécuter une instruction SQL par lots

L'`batch-execute-statement`exemple suivant exécute une instruction SQL par lots sur un tableau de données avec un jeu de paramètres.

```
aws rds-data batch-execute-statement \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --sql "insert into mytable values (:id, :val)" \
    --parameter-sets "[[{\"name\": \"id\", \"value\": {\"longValue\": 1}},{\"name\":
 \"val\", \"value\": {\"stringValue\": \"ValueOne\"}}],
        [{\"name\": \"id\", \"value\": {\"longValue\": 2}},{\"name\": \"val\",
 \"value\": {\"stringValue\": \"ValueTwo\"}}],
        [{\"name\": \"id\", \"value\": {\"longValue\": 3}},{\"name\": \"val\",
 \"value\": {\"stringValue\": \"ValueThree\"}}]]"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de l'API de données pour Aurora Serverless dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous BatchExecuteStatementà la section Référence des AWS CLI commandes.

`begin-transaction`

L'exemple de code suivant montre comment utiliser`begin-transaction`.

AWS CLI

Pour démarrer une transaction SQL

L'`begin-transaction`exemple suivant démarre une transaction SQL.

```
aws rds-data begin-transaction \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret"
```

Sortie :

```
{
    "transactionId": "ABC1234567890xyz"
}
```

Pour plus d'informations, consultez la section Utilisation de l'API de données pour Aurora Serverless dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous BeginTransactionà la section Référence des AWS CLI commandes.

`commit-transaction`

L'exemple de code suivant montre comment utiliser`commit-transaction`.

AWS CLI

Pour valider une transaction SQL

L'`commit-transaction`exemple suivant met fin à la transaction SQL spécifiée et valide les modifications que vous avez apportées dans le cadre de celle-ci.

```
aws rds-data commit-transaction \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --transaction-id "ABC1234567890xyz"
```

Sortie :

```
{
    "transactionStatus": "Transaction Committed"
}
```

Pour plus d'informations, consultez la section Utilisation de l'API de données pour Aurora Serverless dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous CommitTransactionà la section Référence des AWS CLI commandes.

`execute-statement`

L'exemple de code suivant montre comment utiliser`execute-statement`.

AWS CLI

Exemple 1 : pour exécuter une instruction SQL faisant partie d'une transaction

L'`execute-statement`exemple suivant exécute une instruction SQL faisant partie d'une transaction.

```
aws rds-data execute-statement \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --sql "update mytable set quantity=5 where id=201" \
    --transaction-id "ABC1234567890xyz"
```

Sortie :

```
{
    "numberOfRecordsUpdated": 1
```

```
}
```

Exemple 2 : pour exécuter une instruction SQL avec des paramètres

L'`execute-statement`exemple suivant exécute une instruction SQL avec des paramètres.

```
aws rds-data execute-statement \
    --resource-arn "arn:aws:rds:us-east-1:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-east-1:123456789012:secret:mysecret" \
    --sql "insert into mytable values (:id, :val)" \
    --parameters "[{\"name\": \"id\", \"value\": {\"longValue\": 1}},{\"name\":
 \"val\", \"value\": {\"stringValue\": \"value1\"}}]"
```

Sortie :

```
{
    "numberOfRecordsUpdated": 1
}
```

Pour plus d'informations, consultez la section Utilisation de l'API de données pour Aurora Serverless dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous ExecuteStatementà la section Référence des AWS CLI commandes.

`rollback-transaction`

L'exemple de code suivant montre comment utiliser`rollback-transaction`.

AWS CLI

Pour annuler une transaction SQL

L'`rollback-transaction`exemple suivant annule la transaction SQL spécifiée.

```
aws rds-data rollback-transaction \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --transaction-id "ABC1234567890xyz"
```

Sortie :

```
{
    "transactionStatus": "Rollback Complete"
}
```

Pour plus d'informations, consultez la section Utilisation de l'API de données pour Aurora Serverless dans le guide de l'utilisateur Amazon RDS.

- Pour plus de détails sur l'API, reportez-vous RollbackTransactionà la section Référence des AWS CLI commandes.

Exemples d'Amazon RDS Performance Insights utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon RDS Performance Insights.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-performance-analysis-report`

L'exemple de code suivant montre comment utiliser`create-performance-analysis-report`.

AWS CLI

Pour créer un rapport d'analyse des performances

L'`create-performance-analysis-report`exemple suivant crée un rapport d'analyse des performances avec l'heure de début `1682969503` et de fin `1682979503` de la base de données`db-abcdefg123456789`.

```
aws pi create-performance-analysis-report \
```

```
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --start-time 1682969503 \
    --end-time 1682979503
```

Sortie :

```
{
    "AnalysisReportId": "report-0234d3ed98e28fb17"
}
```

Pour plus d'informations sur la création de rapports d'analyse des performances, consultez Création d'un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon RDS et Création d'un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir CreatePerformanceAnalysisReportla section Référence des AWS CLI commandes.

`delete-performance-analysis-report`

L'exemple de code suivant montre comment utiliser`delete-performance-analysis-report`.

AWS CLI

Pour supprimer un rapport d'analyse des performances

L'`delete-performance-analysis-report`exemple suivant supprime le rapport d'analyse des performances avec l'ID `report-0d99cc91c4422ee61` du rapport.

```
aws pi delete-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --analysis-report-id report-0d99cc91c4422ee61
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur la suppression de rapports d'analyse des performances, consultez Supprimer un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon RDS et Suppression d'un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DeletePerformanceAnalysisReportla section Référence des AWS CLI commandes.

`describe-dimension-keys`

L'exemple de code suivant montre comment utiliser`describe-dimension-keys`.

AWS CLI

Exemple 1 : pour décrire les clés de dimension

Cet exemple demande les noms de tous les événements d'attente. Les données sont résumées par nom d'événement et par valeurs agrégées de ces événements sur la période spécifiée.

Commande :

```
aws pi describe-dimension-keys --service-type RDS --identifier db-
LKCGOBK26374TPTDFXOIWVCPPM --start-time 1527026400 --end-time 1527080400 --
metric db.load.avg --group-by '{"Group":"db.wait_event"}'
```

Sortie :

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Keys": [
        {
            "Dimensions": {"db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"},
            "Total": 0.05906906851195666
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/aurora_redo_log_flush"},
            "Total": 0.015824722186149193
        },
        {
            "Dimensions": {"db.wait_event.name": "CPU"},
            "Total": 0.008014396230265477
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/
aurora_respond_to_client"},
```

```
            "Total": 0.0036361612526204477
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/table/sql/handler"},
            "Total": 0.0019108398419382965
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/synch/cond/mysys/
my_thread_var::suspend"},
            "Total": 8.533847837782684E-4
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/file/csv/data"},
            "Total": 6.864181956477376E-4
        },
        {
            "Dimensions": {"db.wait_event.name": "Unknown"},
            "Total": 3.895887056379051E-4
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/synch/mutex/sql/
FILE_AS_TABLE::LOCK_shim_lists"},
            "Total": 3.710368625122906E-5
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/lock/table/sql/handler"},
            "Total": 0
        }
    ]
}
```

Exemple 2 : pour trouver l'ID SQL des instructions qui contribuent le plus à la charge de la base de données

La section suivante `describe-dimension-keys` demande l'instruction SQL et l'ID SQL pour les 10 instructions qui ont le plus contribué à la charge de la base de données.

```
aws pi describe-dimension-keys \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --start-time 2023-05-01T00:00:00Z \
    --end-time 2023-05-01T01:00:00Z \
    --metric db.load.avg \
```

```
    --group-by '{"Group": "db.sql", "Dimensions": ["db.sql.id",
 "db.sql.statement"],"Limit": 10}'
```

Sortie :

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Identifier": "db-abcdefg123456789",
    "MetricList": [
        {
            "Keys": [
                {
                    "Dimensions": {"db.sql.id": "AKIAIOSFODNN7EXAMPLE",
 "db.sql.statement": "SELECT * FROM customers WHERE customer_id = 123"},
                    "Total": 25.5,"Partitions": [12.3, 13.2]
                }
            ]
        }
    ]
}
```

Pour plus d'informations sur les dimensions dans Performance Insights, consultez les sections Chargement de base de données dans le guide de l'utilisateur Amazon RDS et chargement de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir DescribeDimensionKeysla section Référence des AWS CLI commandes.

`get-dimension-key-details`

L'exemple de code suivant montre comment utiliser`get-dimension-key-details`.

AWS CLI

Pour obtenir des détails sur un groupe de dimensions spécifié pour une instance de base de données

L'`get-dimension-key-details`exemple suivant récupère le texte intégral d'une instruction SQL pour une instance `db-10BCD2EFGHIJ3KL4M5NO6PQRS5` de base de données. Le `--group` est `db.sql` et l'`--group-identifier` est `db.sql.id`. Dans cet exemple, `example-sql-id` représente un ID SQL récupéré à l'aide des `describe-dimension-keys` opérations `get-`

`resource-metrics` or. Dans cet exemple, les détails des dimensions sont disponibles. Ainsi, Performance Insights récupère le texte intégral de l'instruction SQL, sans le tronquer.

```
aws pi get-dimension-key-details \
    --service-type RDS \
    --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
    --group db.sql \
    --group-identifier example-sql-id \
    --requested-dimensions statement
```

Sortie :

```
{
    "Dimensions":[
        {
            "Value": "SELECT e.last_name, d.department_name FROM employees e,
 departments d WHERE e.department_id=d.department_id",
            "Dimension": "db.sql.statement",
            "Status": "AVAILABLE"
        },
    ...
    ]
}
```

Pour plus d'informations sur les dimensions dans Performance Insights, consultez les sections Chargement de base de données dans le guide de l'utilisateur Amazon RDS et chargement de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir GetDimensionKeyDetailsla section Référence des AWS CLI commandes.

`get-performance-analysis-report`

L'exemple de code suivant montre comment utiliser`get-performance-analysis-report`.

AWS CLI

Pour obtenir un rapport d'analyse des performances

L'`get-performance-analysis-report`exemple suivant permet d'obtenir le rapport d'analyse des performances de la base de données `db-abcdefg123456789` avec l'ID du

rapportreport-0d99cc91c4422ee61. La réponse fournit le statut du rapport, l'identifiant, les détails temporels et les informations.

```
aws pi get-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --analysis-report-id report-0d99cc91c4422ee61
```

Sortie :

```
{
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-abcdefg123456789",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
            ... (Condensed for space)
       ]
    }
}
```

Pour plus d'informations sur les rapports d'analyse des performances, consultez Analyser les performances des bases de données pendant un certain temps dans le guide de l'utilisateur Amazon RDS et Analyser les performances des bases de données pendant un certain temps dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir GetPerformanceAnalysisReportla section Référence des AWS CLI commandes.

`get-resource-metadata`

L'exemple de code suivant montre comment utiliserget-resource-metadata.

AWS CLI

Pour obtenir les métadonnées des ressources d'une base de données

L'`get-resource-metadata`exemple suivant permet d'obtenir les métadonnées des ressources pour la base de données`db-abcdefg123456789`. La réponse indique que les statistiques SQL Digest sont activées.

```
aws pi get-resource-metadata \
    --service-type RDS \
    --identifier db-abcdefg123456789
```

Sortie :

```
{
    "Identifier": "db-abcdefg123456789",
    "Features":{
        "SQL_DIGEST_STATISTICS":{
            "Status": "ENABLED"
        }
    }
}
```

Pour plus d'informations sur les statistiques SQL pour Performance Insights, consultez les statistiques SQL pour Performance Insights dans le guide de l'utilisateur Amazon RDS et les statistiques SQL pour Performance Insights dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir GetResourceMetadatala section Référence des AWS CLI commandes.

`get-resource-metrics`

L'exemple de code suivant montre comment utiliser`get-resource-metrics`.

AWS CLI

Pour obtenir des indicateurs de ressources

Cet exemple demande des points de données pour le groupe de dimensions db.wait_event et pour la dimension db.wait_event.name au sein de ce groupe. Dans la réponse, les points de données pertinents sont regroupés selon la dimension demandée (db.wait_event.name).

Commande :

```
aws pi get-resource-metrics --service-type RDS --identifier db-
LKCGOBK26374TPTDFXOIWVCPPM --start-time 1527026400 --end-time 1527080400 --period-
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json
```

Les arguments pour `--metric-queries` sont stockés dans un fichier JSON,`metric-queries.json`. Voici le contenu de ce fichier :

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group":"db.wait_event"
        }
    }
]
```

Sortie :

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Identifier": "db-LKCGOBK26374TPTDFXOIWVCPPM",
    "MetricList": [
        {
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": 1527026700.0,
                    "Value": 1.3533333333333333
                },
                {
                    "Timestamp": 1527027000.0,
                    "Value": 0.88
                },
                <...remaining output omitted...>
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
```

```
                "Dimensions": {
                    "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1527026700.0,
                    "Value": 0.8566666666666667
                },
                {
                    "Timestamp": 1527027000.0,
                    "Value": 0.8633333333333333
                },
                <...remaining output omitted...>
            ],
        },
            <...remaining output omitted...>
    ]
}
```

- Pour plus de détails sur l'API, voir GetResourceMetricsla section Référence des AWS CLI commandes.

`list-available-resource-dimensions`

L'exemple de code suivant montre comment utiliser`list-available-resource-dimensions`.

AWS CLI

Pour répertorier les dimensions qui peuvent être demandées pour un type de métrique sur une instance de base de données

L'`list-available-resource-dimensions`exemple suivant répertorie les `db.load` métriques que vous pouvez interroger pour la base de données`db-abcdefg123456789`.

```
aws pi list-available-resource-dimensions \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --metrics db.load
```

Sortie :

```
{
    "MetricDimensions": [
        {
            "Metric": "db.load",
            "Groups": [
                {
                    "Group": "db.user",
                    "Dimensions": [
                        {
                            "Identifier": "db.user.id"
                        },
                        {
                            "Identifier": "db.user.name"
                        }
                    ]
                },
                {
                    "Group": "db.sql_tokenized",
                    "Dimensions": [
                        {
                            "Identifier": "db.sql_tokenized.id"
                        },
                        {
                            "Identifier": "db.sql_tokenized.db_id"
                        },
                        {
                            "Identifier": "db.sql_tokenized.statement"
                        }
                    ]
                },
                ...
            ]
        }
    ]
}
```

Pour plus d'informations sur les dimensions dans Performance Insights, consultez les sections Chargement de base de données dans le guide de l'utilisateur Amazon RDS et chargement de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir ListAvailableResourceDimensionsla section Référence des AWS CLI commandes.

`list-available-resource-metrics`

L'exemple de code suivant montre comment utiliser`list-available-resource-metrics`.

AWS CLI

Pour répertorier les métriques qui peuvent être demandées pour un type de métrique sur une instance de base de données

L'`list-available-resource-metrics`exemple suivant répertorie les `db.load` métriques que vous pouvez interroger pour la base de donnéesdb-abcdefg123456789.

```
aws pi list-available-resource-metrics \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --metric-types "os" "db"
```

Sortie :

```
{
    "Metrics": [
        {
            "Description": "The number of virtual CPUs for the DB instance",
            "Metric": "os.general.numVCPUs",
            "Unit": "vCPUs"
        },
        ......,
        {
            "Description": "Time spent reading data file blocks by backends in this
 instance",
            "Metric": "db.IO.read_latency",
            "Unit": "Milliseconds per block"
        },
        ......
    ]
}
```

Pour plus d'informations sur les métriques dans Performance Insights, consultez les sections Chargement de base de données dans le guide de l'utilisateur Amazon RDS et chargement de base de données dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir ListAvailableResourceMetricsla section Référence des AWS CLI commandes.

`list-performance-analysis-reports`

L'exemple de code suivant montre comment utiliser`list-performance-analysis-reports`.

AWS CLI

Pour répertorier les rapports d'analyse des performances d'une base de données

L'`list-performance-analysis-reports`exemple suivant répertorie les rapports d'analyse des performances de la base de donnéesdb-abcdefg123456789. La réponse répertorie tous les rapports avec l'ID du rapport, le statut et les détails temporels de la période.

```
aws pi list-performance-analysis-reports \
    --service-type RDS \
    --identifier db-abcdefg123456789
```

Sortie :

```
{
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreateTime": 1680587087.139,
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681491137.914,
            "CreateTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreateTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
```

```
            "EndTime": 1682979503.0,
            "CreateTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
        }
    ]
}
```

Pour plus d'informations sur les rapports d'analyse des performances, consultez Analyser les performances des bases de données pendant un certain temps dans le guide de l'utilisateur Amazon RDS et Analyser les performances des bases de données pendant un certain temps dans le guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir ListPerformanceAnalysisReportsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'un rapport d'analyse des performances

L'`list-tags-for-resource`exemple suivant répertorie les balises d'un rapport d'analyse des performances avec l'ID du rapport`report-0d99cc91c4422ee61`.

```
aws pi list-tags-for-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-abcdefg123456789/report-0d99cc91c4422ee61
```

Sortie :

```
{
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

Pour plus d'informations sur le balisage des rapports d'analyse des performances, consultez Ajouter des balises à un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon RDS et Ajout de balises à un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une balise à un rapport d'analyse des performances

L'`tag-resource`exemple suivant ajoute la clé de balise `name` avec la valeur de la balise `test-tag` à un rapport d'analyse des performances avec l'ID du rapport`report-0d99cc91c4422ee61`.

```
aws pi tag-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-abcdefg123456789/report-0d99cc91c4422ee61 \
    --tags Key=name,Value=test-tag
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur le balisage des rapports d'analyse des performances, consultez Ajouter des balises à un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon RDS et Ajout de balises à un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une balise pour un rapport d'analyse des performances

L'`untag-resource`exemple suivant supprime la balise d'un rapport `name` d'analyse des performances avec l'ID `report-0d99cc91c4422ee61` du rapport.

```
aws pi untag-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-abcdefg123456789/report-0d99cc91c4422ee61 \
    --tag-keys name
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur le balisage des rapports d'analyse des performances, consultez Ajouter des balises à un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon RDS et Ajout de balises à un rapport d'analyse des performances dans Performance Insights du guide de l'utilisateur Amazon Aurora.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Redshift AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Redshift.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-reserved-node-exchange`

L'exemple de code suivant montre comment utiliser`accept-reserved-node-exchange`.

AWS CLI

Pour accepter l'échange de nœuds réservés

L'`accept-reserved-node-exchange`exemple suivant accepte l'échange d'un nœud DC1 réservé contre un nœud DC2 réservé.

```
aws redshift accept-reserved-node-exchange /
    --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE /
    --target-reserved-node-offering-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

Sortie :

```
{
    "ExchangedReservedNode": {
        "ReservedNodeId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
        "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
        "NodeType": "dc2.large",
        "StartTime": "2019-12-06T21:17:26Z",
        "Duration": 31536000,
        "FixedPrice": 0.0,
        "UsagePrice": 0.0,
        "CurrencyCode": "USD",
        "NodeCount": 1,
        "State": "exchanging",
        "OfferingType": "All Upfront",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.0,
                "RecurringChargeFrequency": "Hourly"
            }
        ],
        "ReservedNodeOfferingType": "Regular"
    }
}
```

Pour plus d'informations, consultez la section Mise à niveau des nœuds réservés avec la AWS CLI dans le manuel Amazon Redshift Cluster Management Guide.

- Pour plus de détails sur l'API, voir AcceptReservedNodeExchangela section Référence des AWS CLI commandes.

`authorize-cluster-security-group-ingress`

L'exemple de code suivant montre comment utiliser`authorize-cluster-security-group-ingress`.

AWS CLI

L' GroupThis exemple d'autorisation de l'accès à un EC2 groupe de sécurité autorise l'accès à un groupe de EC2 sécurité Amazon nommé.Commande :

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

Autorisation de l'accès à une plage d'adresses CIDRCet exemple autorise l'accès à une plage d'adresses CIDR.Commande :

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- Pour plus de détails sur l'API, voir AuthorizeClusterSecurityGroupIngressla section Référence des AWS CLI commandes.

`authorize-snapshot-access`

L'exemple de code suivant montre comment utiliser`authorize-snapshot-access`.

AWS CLI

Autoriser un AWS compte pour restaurer Un SnapshotThis exemple autorise le AWS compte `444455556666` à restaurer le snapshot`my-snapshot-id`. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

Résultat:

```
{
   "Snapshot": {
      "Status": "available",
      "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
      "EstimatedSecondsToCompletion": 0,
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "Encrypted": false,
      "OwnerAccount": "111122223333",
      "BackupProgressInMegabytes": 11.0,
      "ElapsedTimeInSeconds": 0,
      "DBName": "dev",
      "CurrentBackupRateInMegabytesPerSecond: 0.1534,
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "ActualIncrementalBackupSizeInMegabytes"; 11.0,
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "TotalBackupSizeInMegabytes": 20.0,
      "Port": 5439,
      "NumberOfNodes": 2,
      "SnapshotIdentifier": "my-snapshot-id"
   }
}
```

- Pour plus de détails sur l'API, voir AuthorizeSnapshotAccessla section Référence des AWS CLI commandes.

`batch-delete-cluster-snapshots`

L'exemple de code suivant montre comment utiliser`batch-delete-cluster-snapshots`.

AWS CLI

Pour supprimer un ensemble de snapshots de cluster

L'`batch-delete-cluster-snapshots`exemple suivant supprime un ensemble de clichés de cluster manuels.

```
aws redshift batch-delete-cluster-snapshots \
        --
identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-
```

Sortie :

```
{
    "Resources": [
        "mycluster-2019-11-06-14-12",
        "mycluster-2019-11-06-14-20"
    ]
}
```

Pour plus d'informations, consultez Amazon Redshift Snapshots dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir BatchDeleteClusterSnapshotsla section Référence des AWS CLI commandes.

`batch-modify-cluster-snapshots`

L'exemple de code suivant montre comment utiliser`batch-modify-cluster-snapshots`.

AWS CLI

Pour modifier un ensemble d'instantanés de cluster

L'`batch-modify-cluster-snapshots`exemple suivant modifie les paramètres d'un ensemble de clichés de cluster.

```
aws redshift batch-modify-cluster-snapshots \
    --snapshot-identifier-list mycluster-2019-11-06-16-31 mycluster-2019-11-06-16-32
 \
    --manual-snapshot-retention-period 30
```

Sortie :

```
{
    "Resources": [
```

```
        "mycluster-2019-11-06-16-31",
        "mycluster-2019-11-06-16-32"
    ],
    "Errors": [],
    "ResponseMetadata": {
        "RequestId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
                "x-amzn-requestid": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE,
                "content-type": "text/xml",
                "content-length": "480",
                "date": "Sat, 07 Dec 2019 00:36:09 GMT",
                "connection": "keep-alive"
        },
        "RetryAttempts": 0
    }
}
```

Pour plus d'informations, consultez Amazon Redshift Snapshots dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir BatchModifyClusterSnapshotsla section Référence des AWS CLI commandes.

`cancel-resize`

L'exemple de code suivant montre comment utiliser`cancel-resize`.

AWS CLI

Pour annuler le redimensionnement d'un cluster

L'`cancel-resize`exemple suivant annule une opération de redimensionnement classique pour un cluster.

```
aws redshift cancel-resize \
    --cluster-identifier mycluster
```

Sortie :

```
{
    "TargetNodeType": "dc2.large",
    "TargetNumberOfNodes": 2,
```

```
    "TargetClusterType": "multi-node",
    "Status": "CANCELLING",
    "ResizeType": "ClassicResize",
    "TargetEncryptionType": "NONE"
}
```

Pour plus d'informations, consultez la section Redimensionnement des clusters dans Amazon Redshift dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir CancelResizela section Référence des AWS CLI commandes.

`copy-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`copy-cluster-snapshot`.

AWS CLI

L' VersionsThis exemple Get a Description of All Cluster renvoie une description de toutes les versions du cluster. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
 cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

Résultat:

```
{
   "Snapshot": {
      "Status": "available",
      "SnapshotCreateTime": "2013-01-22T19:27:58.931Z",
      "AvailabilityZone": "us-east-1c",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "DBName": "dev",
      "ClusterCreateTime": "2013-01-22T19:23:59.368Z",
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "examplecluster",
      "Port": 5439,
      "NumberOfNodes": "2",
      "SnapshotIdentifier": "my-saved-snapshot-copy"
```

```
    },
    "ResponseMetadata": {
        "RequestId": "3b279691-64e3-11e2-bec0-17624ad140dd"
    }
}
```

- Pour plus de détails sur l'API, voir CopyClusterSnapshotla section Référence des AWS CLI commandes.

`create-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`create-cluster-parameter-group`.

AWS CLI

L' GroupThis exemple de création d'un paramètre de cluster crée un nouveau groupe de paramètres de cluster.Commande :

```
aws redshift create-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
 first cluster parameter group"
```

Résultat:

```
{
    "ClusterParameterGroup": {
        "ParameterGroupFamily": "redshift-1.0",
        "Description": "My first cluster parameter group",
        "ParameterGroupName": "myclusterparametergroup"
    },
    "ResponseMetadata": {
        "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
    }
}
```

- Pour plus de détails sur l'API, voir CreateClusterParameterGroupla section Référence des AWS CLI commandes.

`create-cluster-security-group`

L'exemple de code suivant montre comment utiliser`create-cluster-security-group`.

AWS CLI

La création d'un GroupThis exemple de sécurité de cluster crée un nouveau groupe de sécurité de cluster. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift create-cluster-security-group --cluster-security-group-name
 mysecuritygroup --description "This is my cluster security group"
```

Résultat:

```
{
   "create_cluster_security_group_response": {
      "create_cluster_security_group_result": {
         "cluster_security_group": {
            "description": "This is my cluster security group",
            "owner_id": "300454760768",
            "cluster_security_group_name": "mysecuritygroup",
            "ec2_security_groups": \[],
            "ip_ranges": \[]
         }
      },
      "response_metadata": {
         "request_id": "5df486a0-343a-11e2-b0d8-d15d0ef48549"
      }
   }
}
```

Vous pouvez également obtenir les mêmes informations au format texte à l'aide de l'`--output text`option .Commande :

`--output text`Option. Commande :

Option. Commande :

```
aws redshift create-cluster-security-group --cluster-security-group-name
 mysecuritygroup --description "This is my cluster security group" --output text
```

Résultat:

```
This is my cluster security group   300454760768    mysecuritygroup
```

```
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- Pour plus de détails sur l'API, voir CreateClusterSecurityGroupla section Référence des AWS CLI commandes.

`create-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`create-cluster-snapshot`.

AWS CLI

L' SnapshotThis exemple de création d'un cluster crée un nouvel instantané de cluster. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-
identifier my-snapshot-id
```

Résultat:

```
{
   "Snapshot": {
      "Status": "creating",
      "SnapshotCreateTime": "2013-01-22T22:20:33.548Z",
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "DBName": "dev",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "Port": 5439,
      "NumberOfNodes": "2",
      "SnapshotIdentifier": "my-snapshot-id"
   },
   "ResponseMetadata": {
      "RequestId": "f024d1a5-64e1-11e2-88c5-53eb05787dfb"
   }
}
```

- Pour plus de détails sur l'API, voir CreateClusterSnapshotla section Référence des AWS CLI commandes.

`create-cluster-subnet-group`

L'exemple de code suivant montre comment utiliser`create-cluster-subnet-group`.

AWS CLI

L' GroupThis exemple de création d'un sous-réseau de cluster crée un nouveau groupe de sous-réseaux de cluster.Commande :

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
 --description "My subnet group" --subnet-ids subnet-763fdd1c
```

Résultat:

```
{
   "ClusterSubnetGroup": {
      "Subnets": [
         {
            "SubnetStatus": "Active",
            "SubnetIdentifier": "subnet-763fdd1c",
            "SubnetAvailabilityZone": {
               "Name": "us-east-1a"
            }
         } ],
      "VpcId": "vpc-7e3fdd14",
      "SubnetGroupStatus": "Complete",
      "Description": "My subnet group",
      "ClusterSubnetGroupName": "mysubnetgroup"
   },
   "ResponseMetadata": {
      "RequestId": "500b8ce2-698f-11e2-9790-fd67517fb6fd"
   }
}
```

- Pour plus de détails sur l'API, voir CreateClusterSubnetGroupla section Référence des AWS CLI commandes.

`create-cluster`

L'exemple de code suivant montre comment utiliser`create-cluster`.

AWS CLI

L' ParametersThis exemple Create a Cluster with Minimal crée un cluster avec un ensemble minimal de paramètres. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-
username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

Résultat:

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "default.redshift-1.0"
         } ],
      "ClusterSecurityGroups": [
         {
            "Status": "active",
            "ClusterSecurityGroupName": "default"
         } ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "creating",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {
         "MasterUserPassword": "\****"
      }
   },
   "ResponseMetadata": {
      "RequestId": "7cf4bcfc-64dd-11e2-bea9-49e0ce183f07"
   }
}
```

- Pour plus de détails sur l'API, voir CreateClusterla section Référence des AWS CLI commandes.

`create-event-subscription`

L'exemple de code suivant montre comment utiliser`create-event-subscription`.

AWS CLI

Pour créer un abonnement aux notifications pour un événement

L'`create-event-subscription`exemple suivant crée un abonnement aux notifications d'événements.

```
aws redshift create-event-subscription \
    --subscription-name mysubscription \
    --sns-topic-arn arn:aws:sns:us-west-2:123456789012:MySNStopic \
    --source-type cluster \
    --source-ids mycluster
```

Sortie :

```
{
        "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "mysubscription",
        "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
        "Status": "active",
        "SubscriptionCreationTime": "2019-12-09T20:05:19.365Z",
        "SourceType": "cluster",
        "SourceIdsList": [
            "mycluster"
        ],
        "EventCategoriesList": [],
        "Severity": "INFO",
        "Enabled": true,
        "Tags": []
    }
}
```

Pour plus d'informations, consultez la section Abonnement aux notifications d'événements Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir CreateEventSubscriptionla section Référence des AWS CLI commandes.

`create-hsm-client-certificate`

L'exemple de code suivant montre comment utiliser`create-hsm-client-certificate`.

AWS CLI

Pour créer un certificat client HSM

L'`create-hsm-client-certificate`exemple suivant génère un certificat client HSM qu'un cluster peut utiliser pour se connecter à un HSM.

```
aws redshift create-hsm-client-certificate \
    --hsm-client-certificate-identifier myhsmclientcert
```

Sortie :

```
{
    "HsmClientCertificate": {
        "HsmClientCertificateIdentifier": "myhsmclientcert",
        "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----
        MIICiEXAMPLECQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
        VVMxCzAJBgNVBAgTEXAMPLEwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
        b24xFDASBgNVBAsTC0lBTSBDb25EXAMPLEIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
        BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN
        MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD
        EXAMPLETZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
        b2xlMRIwEAEXAMPLEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
        YXpvbi5jb20wgZ8wDQYJKEXAMPLEAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
        21uUSfwfEvySWtC2XADZ4nB+BLYgVIk6EXAMPLE3G93vUEIO3IyNoH/f0wYK8m9T
        rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugEXAMPLEzZswY6786m86gpE
        Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEXAMPLEEAtCu4
        nUhVVxYUEXAMPLEh8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
        FFBjvSfpJIlJ00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
        NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
    }
}
```

Pour plus d'informations, consultez la référence des autorisations d'API Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir CreateHsmClientCertificatela section Référence des AWS CLI commandes.

`create-hsm-configuration`

L'exemple de code suivant montre comment utiliser`create-hsm-configuration`.

AWS CLI

Pour créer une configuration HSM

L'`create-hsm-configuration`exemple suivant crée la configuration HSM spécifiée qui contient les informations requises par un cluster pour stocker et utiliser les clés de chiffrement de base de données dans un module de sécurité matériel (HSM).

```
aws redshift create-hsm-configuration /
    --hsm-configuration-identifier myhsmconnection
    --description "My HSM connection"
    --hsm-ip-address 192.0.2.09
    --hsm-partition-name myhsmpartition /
    --hsm-partition-password A1b2c3d4 /
    --hsm-server-public-certificate myhsmclientcert
```

Sortie :

```
{
    "HsmConfiguration": {
        "HsmConfigurationIdentifier": "myhsmconnection",
        "Description": "My HSM connection",
        "HsmIpAddress": "192.0.2.09",
        "HsmPartitionName": "myhsmpartition",
        "Tags": []
    }
}
```

- Pour plus de détails sur l'API, voir CreateHsmConfigurationla section Référence des AWS CLI commandes.

`create-snapshot-copy-grant`

L'exemple de code suivant montre comment utiliser`create-snapshot-copy-grant`.

AWS CLI

Pour créer une autorisation de copie instantanée

L'`create-snapshot-copy-grant`exemple suivant crée une autorisation de copie d'instantanés et chiffre les instantanés copiés dans une région de destination AWS .

```
aws redshift create-snapshot-copy-grant \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

Sortie :

```
{
    "SnapshotCopyGrant": {
        "SnapshotCopyGrantName": "mysnapshotcopygrantname",
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
        "Tags": []
    }
}
```

Pour plus d'informations, consultez Amazon Redshift Database Encryption dans le manuel Amazon Redshift Cluster Management Guide.

- Pour plus de détails sur l'API, voir CreateSnapshotCopyGrantla section Référence des AWS CLI commandes.

`create-snapshot-schedule`

L'exemple de code suivant montre comment utiliser`create-snapshot-schedule`.

AWS CLI

Pour créer un calendrier d'instantanés

L'`create-snapshot-schedule`exemple suivant crée un calendrier de captures d'écran avec la description spécifiée et une fréquence toutes les 12 heures.

```
aws redshift create-snapshot-schedule \
    --schedule-definitions "rate(12 hours)" \
    --schedule-identifier mysnapshotschedule \
    --schedule-description "My schedule description"
```

Sortie :

```
{
    "ScheduleDefinitions": [
        "rate(12 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
}
```

Pour plus d'informations, consultez la section Planification automatisée des instantanés dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir CreateSnapshotSchedulela section Référence des AWS CLI commandes.

`create-tags`

L'exemple de code suivant montre comment utiliser`create-tags`.

AWS CLI

Pour créer des balises pour un cluster

L'`create-tags`exemple suivant ajoute la key/value paire de balises spécifiée au cluster spécifié.

```
aws redshift create-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tags "Key"="mytags","Value"="tag1"
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Ressources de balisage dans Amazon Redshift dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir CreateTagsla section Référence des AWS CLI commandes.

`delete-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`delete-cluster-parameter-group`.

AWS CLI

L' GroupThis exemple Supprimer un paramètre de cluster supprime un groupe de paramètres de cluster.Commande :

```
aws redshift delete-cluster-parameter-group --parameter-group-name
 myclusterparametergroup
```

- Pour plus de détails sur l'API, voir DeleteClusterParameterGroupla section Référence des AWS CLI commandes.

`delete-cluster-security-group`

L'exemple de code suivant montre comment utiliser`delete-cluster-security-group`.

AWS CLI

Supprimer un GroupThis exemple de sécurité de cluster supprime un groupe de sécurité de cluster.Commande :

```
aws redshift delete-cluster-security-group --cluster-security-group-name
 mysecuritygroup
```

- Pour plus de détails sur l'API, voir DeleteClusterSecurityGroupla section Référence des AWS CLI commandes.

`delete-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`delete-cluster-snapshot`.

AWS CLI

Supprimer un cluster SnapshotThis Exemple de suppression d'un instantané de cluster.Commande :

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- Pour plus de détails sur l'API, voir DeleteClusterSnapshotla section Référence des AWS CLI commandes.

`delete-cluster-subnet-group`

L'exemple de code suivant montre comment utiliser`delete-cluster-subnet-group`.

AWS CLI

Supprimer un sous-réseau de cluster. GroupThis Exemple de suppression d'un groupe de sous-réseaux de cluster.Commande :

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

Résultat:

```
{
   "ResponseMetadata": {
      "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
   }
}
```

- Pour plus de détails sur l'API, voir DeleteClusterSubnetGroupla section Référence des AWS CLI commandes.

`delete-cluster`

L'exemple de code suivant montre comment utiliser`delete-cluster`.

AWS CLI

Supprimer un cluster sans cluster final SnapshotThis Exemple de suppression d'un cluster, forçant ainsi la suppression des données afin qu'aucun instantané final du cluster ne soit créé.Commande :

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-
snapshot
```

Supprimer un cluster, autoriser un cluster final, SnapshotThis par exemple, supprime un cluster, mais spécifie un instantané du cluster.Commande :

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- Pour plus de détails sur l'API, voir DeleteClusterla section Référence des AWS CLI commandes.

`delete-event-subscription`

L'exemple de code suivant montre comment utiliser`delete-event-subscription`.

AWS CLI

Pour supprimer l'abonnement à un événement

L'`delete-event-subscription`exemple suivant supprime l'abonnement aux notifications d'événements spécifié.

```
aws redshift delete-event-subscription \
    --subscription-name mysubscription
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Abonnement aux notifications d'événements Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DeleteEventSubscriptionla section Référence des AWS CLI commandes.

`delete-hsm-client-certificate`

L'exemple de code suivant montre comment utiliser`delete-hsm-client-certificate`.

AWS CLI

Pour supprimer le certificat client HSM

L'`delete-hsm-client-certificate`exemple suivant supprime un certificat client HSM.

```
aws redshift delete-hsm-client-certificate \
    --hsm-client-certificate-identifier myhsmclientcert
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la référence des autorisations d'API Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DeleteHsmClientCertificatela section Référence des AWS CLI commandes.

`delete-hsm-configuration`

L'exemple de code suivant montre comment utiliser`delete-hsm-configuration`.

AWS CLI

Pour supprimer une configuration HSM

L'`delete-hsm-configuration`exemple suivant supprime la configuration HSM spécifiée du compte actuel AWS .

```
aws redshift delete-hsm-configuration /
    --hsm-configuration-identifier myhsmconnection
```

Cette commande ne génère pas de sortie.

- Pour plus de détails sur l'API, voir DeleteHsmConfigurationla section Référence des AWS CLI commandes.

`delete-scheduled-action`

L'exemple de code suivant montre comment utiliser`delete-scheduled-action`.

AWS CLI

Pour supprimer une action planifiée

L'`delete-scheduled-action`exemple suivant supprime l'action planifiée spécifiée.

```
aws redshift delete-scheduled-action \
```

```
    --scheduled-action-name myscheduledaction
```

Cette commande ne génère pas de sortie.

- Pour plus de détails sur l'API, voir DeleteScheduledActionla section Référence des AWS CLI commandes.

`delete-snapshot-copy-grant`

L'exemple de code suivant montre comment utiliser`delete-snapshot-copy-grant`.

AWS CLI

Pour supprimer l'autorisation de copie instantanée

L'`delete-snapshot-copy-grant`exemple suivant supprime l'autorisation de copie d'instantané spécifiée.

```
aws redshift delete-snapshot-copy-grant \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez Amazon Redshift Database Encryption dans le manuel Amazon Redshift Cluster Management Guide.

- Pour plus de détails sur l'API, voir DeleteSnapshotCopyGrantla section Référence des AWS CLI commandes.

`delete-snapshot-schedule`

L'exemple de code suivant montre comment utiliser`delete-snapshot-schedule`.

AWS CLI

Pour supprimer le calendrier des instantanés

L'`delete-snapshot-schedule`exemple suivant supprime le calendrier de capture d'écran spécifié. Vous devez dissocier les clusters avant de supprimer le calendrier.

```
aws redshift delete-snapshot-schedule \
```

```
    --schedule-identifier mysnapshotschedule
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Planification automatisée des instantanés dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DeleteSnapshotSchedulela section Référence des AWS CLI commandes.

`delete-tags`

L'exemple de code suivant montre comment utiliser`delete-tags`.

AWS CLI

Pour supprimer des balises d'un cluster

L'`delete-tags`exemple suivant supprime les balises portant les noms de clé spécifiés du cluster spécifié.

```
aws redshift delete-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tag-keys "clustertagkey" "clustertagvalue"
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Ressources de balisage dans Amazon Redshift dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DeleteTagsla section Référence des AWS CLI commandes.

`describe-account-attributes`

L'exemple de code suivant montre comment utiliser`describe-account-attributes`.

AWS CLI

Pour décrire les attributs d'un AWS compte

L'`describe-account-attributes`exemple suivant affiche les attributs associés au AWS compte d'appel.

```
aws redshift describe-account-attributes
```

Sortie :

```
{
    "AccountAttributes": [
        {
            "AttributeName": "max-defer-maintenance-duration",
            "AttributeValues": [
                {
                    "AttributeValue": "45"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeAccountAttributesla section Référence des AWS CLI commandes.

`describe-cluster-db-revisions`

L'exemple de code suivant montre comment utiliser`describe-cluster-db-revisions`.

AWS CLI

Pour décrire les révisions de base de données pour un cluster

L'`describe-cluster-db-revisions`exemple suivant affiche les détails d'un tableau d'`ClusterDbRevision`objets pour le cluster spécifié.

```
aws redshift describe-cluster-db-revisions \
    --cluster-identifier mycluster
```

Sortie :

```
{
    "ClusterDbRevisions": [
        {
            "ClusterIdentifier": "mycluster",
            "CurrentDatabaseRevision": "11420",
```

```
            "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",
            "RevisionTargets": []
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeClusterDbRevisionsla section Référence des AWS CLI commandes.

`describe-cluster-parameter-groups`

L'exemple de code suivant montre comment utiliser`describe-cluster-parameter-groups`.

AWS CLI

L' GroupsThis exemple Obtenir une description de tous les paramètres de cluster renvoie une description de tous les groupes de paramètres de cluster pour le compte, avec des en-têtes de colonne. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-cluster-parameter-groups
```

Résultat:

```
{
   "ParameterGroups": [
      {
         "ParameterGroupFamily": "redshift-1.0",
         "Description": "My first cluster parameter group",
         "ParameterGroupName": "myclusterparametergroup"
      } ],
   "ResponseMetadata": {
      "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
   }
}
```

Vous pouvez également obtenir les mêmes informations au format texte à l'aide de l'`--output text`option .Commande :

`--output text`Option. Commande :

Option. Commande :

```
aws redshift describe-cluster-parameter-groups --output text
```

Résultat:

```
redshift-1.0        My first cluster parameter group        myclusterparametergroup
RESPONSEMETADATA    9e665a36-64cc-11e2-8f7d-3b939af52818
```

- Pour plus de détails sur l'API, voir DescribeClusterParameterGroupsla section Référence des AWS CLI commandes.

`describe-cluster-parameters`

L'exemple de code suivant montre comment utiliser`describe-cluster-parameters`.

AWS CLI

L' GroupThis exemple de récupération des paramètres d'un paramètre de cluster spécifié permet de récupérer les paramètres du groupe de paramètres nommé. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-cluster-parameters --parameter-group-name
 myclusterparametergroup
```

Résultat:

```
{
   "Parameters": [
      {
         "Description": "Sets the display format for date and time values.",
         "DataType": "string",
         "IsModifiable": true,
         "Source": "engine-default",
         "ParameterValue": "ISO, MDY",
         "ParameterName": "datestyle"
      },
      {
         "Description": "Sets the number of digits displayed for floating-point
 values",
         "DataType": "integer",
         "IsModifiable": true,
         "AllowedValues": "-15-2",
```

```
            "Source": "engine-default",
            "ParameterValue": "0",
            "ParameterName": "extra_float_digits"
        },
        (...remaining output omitted...)
    ]
}
```

Vous pouvez également obtenir les mêmes informations au format texte à l'aide de l'`--output text`option .Commande :

`--output text`Option. Commande :

Option. Commande :

```
aws redshift describe-cluster-parameters --parameter-group-name
 myclusterparametergroup --output text
```

Résultat:

```
RESPONSEMETADATA    cdac40aa-64cc-11e2-9e70-918437dd236d
Sets the display format for date and time values.   string  True    engine-default
 ISO, MDY        datestyle
Sets the number of digits displayed for floating-point values       integer True
 -15-2   engine-default  0       extra_float_digits
This parameter applies a user-defined label to a group of queries that are run
 during the same session..    string  True    engine-default  default query_group
require ssl for all databaseconnections     boolean True    true,false      engine-
default  false   require_ssl
Sets the schema search order for names that are not schema-qualified.       string
 True    engine-default  $user, public   search_path
Aborts any statement that takes over the specified number of milliseconds.  integer
 True    engine-default  0       statement_timeout
wlm json configuration      string  True    engine-default
 \[{"query_concurrency":5}]      wlm_json_configuration
```

- Pour plus de détails sur l'API, voir DescribeClusterParameterslа section Référence des AWS CLI commandes.

`describe-cluster-security-groups`

L'exemple de code suivant montre comment utiliser`describe-cluster-security-groups`.

AWS CLI

L' GroupsThis exemple Get a Description of All Cluster Security renvoie une description de tous les groupes de sécurité du cluster pour le compte. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-cluster-security-groups
```

Résultat:

```
{
   "ClusterSecurityGroups": [
      {
         "OwnerId": "100447751468",
         "Description": "default",
         "ClusterSecurityGroupName": "default",
         "EC2SecurityGroups": \[],
         "IPRanges": [
            {
               "Status": "authorized",
               "CIDRIP": "0.0.0.0/0"
            }
         ]
      },
      {
         "OwnerId": "100447751468",
         "Description": "This is my cluster security group",
         "ClusterSecurityGroupName": "mysecuritygroup",
         "EC2SecurityGroups": \[],
         "IPRanges": \[]
      },
      (...remaining output omitted...)
   ]
}
```

- Pour plus de détails sur l'API, voir DescribeClusterSecurityGroupsla section Référence des AWS CLI commandes.

`describe-cluster-snapshots`

L'exemple de code suivant montre comment utiliser`describe-cluster-snapshots`.

AWS CLI

L' SnapshotsThis exemple Get a Description of All Cluster renvoie une description de tous les instantanés du cluster pour le compte. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-cluster-snapshots
```

Résultat:

```
{
   "Snapshots": [
      {
         "Status": "available",
         "SnapshotCreateTime": "2013-07-17T22:02:22.852Z",
         "EstimatedSecondsToCompletion": -1,
         "AvailabilityZone": "us-east-1a",
         "ClusterVersion": "1.0",
         "MasterUsername": "adminuser",
         "Encrypted": false,
         "OwnerAccount": "111122223333",
         "BackupProgressInMegabytes": 20.0,
         "ElapsedTimeInSeconds": 0,
         "DBName": "dev",
         "CurrentBackupRateInMegabytesPerSecond: 0.0,
         "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
         "ActualIncrementalBackupSizeInMegabytes"; 20.0
         "SnapshotType": "automated",
         "NodeType": "dw.hs1.xlarge",
         "ClusterIdentifier": "mycluster",
         "Port": 5439,
         "TotalBackupSizeInMegabytes": 20.0,
         "NumberOfNodes": "2",
         "SnapshotIdentifier": "cm:mycluster-2013-01-22-22-04-18"
      },
      {
         "EstimatedSecondsToCompletion": 0,
         "OwnerAccount": "111122223333",
         "CurrentBackupRateInMegabytesPerSecond: 0.1534,
         "ActualIncrementalBackupSizeInMegabytes"; 11.0,
         "NumberOfNodes": "2",
         "Status": "available",
         "ClusterVersion": "1.0",
         "MasterUsername": "adminuser",
```

```
         "AccountsWithRestoreAccess": [
            {
               "AccountID": "444455556666"
            } ],
         "TotalBackupSizeInMegabytes": 20.0,
         "DBName": "dev",
         "BackupProgressInMegabytes": 11.0,
         "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
         "ElapsedTimeInSeconds": 0,
         "ClusterIdentifier": "mycluster",
         "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
         "AvailabilityZone": "us-east-1a",
         "NodeType": "dw.hs1.xlarge",
         "Encrypted": false,
         "SnapshotType": "manual",
         "Port": 5439,
         "SnapshotIdentifier": "my-snapshot-id"
      } ]
   }
   (...remaining output omitted...)
```

- Pour plus de détails sur l'API, voir DescribeClusterSnapshotsla section Référence des AWS CLI commandes.

`describe-cluster-subnet-groups`

L'exemple de code suivant montre comment utiliser`describe-cluster-subnet-groups`.

AWS CLI

L' GroupsThis exemple Obtenir une description de tous les sous-réseaux du cluster renvoie une description de tous les groupes de sous-réseaux du cluster. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-cluster-subnet-groups
```

Résultat:

```
{
   "ClusterSubnetGroups": [
      {
```

```
            "Subnets": [
               {
                  "SubnetStatus": "Active",
                  "SubnetIdentifier": "subnet-763fdd1c",
                  "SubnetAvailabilityZone": {
                     "Name": "us-east-1a"
                  }
               }
            ],
            "VpcId": "vpc-7e3fdd14",
            "SubnetGroupStatus": "Complete",
            "Description": "My subnet group",
            "ClusterSubnetGroupName": "mysubnetgroup"
         }
      ],
      "ResponseMetadata": {
         "RequestId": "37fa8c89-6990-11e2-8f75-ab4018764c77"
      }
}
```

- Pour plus de détails sur l'API, voir DescribeClusterSubnetGroupsla section Référence des AWS CLI commandes.

`describe-cluster-tracks`

L'exemple de code suivant montre comment utiliser`describe-cluster-tracks`.

AWS CLI

Pour décrire les pistes du cluster

L'`describe-cluster-tracks`exemple suivant affiche les détails des pistes de maintenance disponibles.

```
aws redshift describe-cluster-tracks \
    --maintenance-track-name current
```

Sortie :

```
{
    "MaintenanceTracks": [
        {
```

```
            "MaintenanceTrackName": "current",
            "DatabaseVersion": "1.0.11420",
            "UpdateTargets": [
                {
                    "MaintenanceTrackName": "preview_features",
                    "DatabaseVersion": "1.0.11746",
                    "SupportedOperations": [
                        {
                            "OperationName": "restore-from-cluster-snapshot"
                        }
                    ]
                },
                {
                    "MaintenanceTrackName": "trailing",
                    "DatabaseVersion": "1.0.11116",
                    "SupportedOperations": [
                        {
                            "OperationName": "restore-from-cluster-snapshot"
                        },
                        {
                            "OperationName": "modify-cluster"
                        }
                    ]
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Choisir les pistes de maintenance des clusters dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DescribeClusterTracksla section Référence des AWS CLI commandes.

`describe-cluster-versions`

L'exemple de code suivant montre comment utiliser`describe-cluster-versions`.

AWS CLI

L' VersionsThis exemple Get a Description of All Cluster renvoie une description de toutes les versions du cluster. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-cluster-versions
```

Résultat:

```
{
   "ClusterVersions": [
      {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
      } ],
   "ResponseMetadata": {
      "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
   }
}
```

- Pour plus de détails sur l'API, voir DescribeClusterVersionsla section Référence des AWS CLI commandes.

`describe-clusters`

L'exemple de code suivant montre comment utiliser`describe-clusters`.

AWS CLI

L' ClustersThis exemple Get a Description of All renvoie une description de tous les clusters du compte. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-clusters
```

Résultat:

```
{
   "Clusters": [
   {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
         "Port": 5439,
         "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
```

```
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "default.redshift-1.0"
         } ],
      "ClusterSecurityGroups": [
         {
            "Status": "active",
            "ClusterSecurityGroupName": "default"
         } ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "AvailabilityZone": "us-east-1a",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "available",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {}
   } ],
   "ResponseMetadata": {
      "RequestId": "65b71cac-64df-11e2-8f5b-e90bd6c77476"
   }
}
```

Vous pouvez également obtenir les mêmes informations au format texte à l'aide de l'`--output text`option .Commande :

`--output text`Option. Commande :

Option. Commande :

```
aws redshift describe-clusters --output text
```

Résultat:

```
dw.hs1.xlarge       1.0     true    adminuser       True    us-east-1a
 2013-01-22T21:59:29.559Z        sat:03:30-sat:04:00     1       available
 mycluster       dev     2
```

```
ENDPOINT    5439    mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com
in-sync     default.redshift-1.0
active      default
PENDINGMODIFIEDVALUES
RESPONSEMETADATA    934281a8-64df-11e2-b07c-f7fbdd006c67
```

- Pour plus de détails sur l'API, voir DescribeClustersla section Référence des AWS CLI commandes.

`describe-default-cluster-parameters`

L'exemple de code suivant montre comment utiliser`describe-default-cluster-parameters`.

AWS CLI

L' ParametersThis exemple Get a Description of Default Cluster renvoie une description des paramètres de cluster par défaut pour la `redshift-1.0` famille. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-default-cluster-parameters --parameter-group-family
 redshift-1.0
```

Résultat:

```
{
   "DefaultClusterParameters": {
   "ParameterGroupFamily": "redshift-1.0",
   "Parameters": [
      {
         "Description": "Sets the display format for date and time values.",
         "DataType": "string",
         "IsModifiable": true,
         "Source": "engine-default",
         "ParameterValue": "ISO, MDY",
         "ParameterName": "datestyle"
      },
      {
         "Description": "Sets the number of digits displayed for floating-point
 values",
         "DataType": "integer",
         "IsModifiable": true,
         "AllowedValues": "-15-2",
```

```
            "Source": "engine-default",
            "ParameterValue": "0",
            "ParameterName": "extra_float_digits"
        },
        (...remaining output omitted...)
        ]
    }
}
```

Pour consulter la liste des familles de groupes de paramètres valides, utilisez la `describe-cluster-parameter-groups` commande.

`describe-cluster-parameter-groups`commande.

commande.

- Pour plus de détails sur l'API, voir DescribeDefaultClusterParameterslа section Référence des AWS CLI commandes.

`describe-event-categories`

L'exemple de code suivant montre comment utiliser`describe-event-categories`.

AWS CLI

Pour décrire les catégories d'événements d'un cluster

L'`describe-event-categories`exemple suivant affiche les détails des catégories d'événements d'un cluster.

```
aws redshift describe-event-categories \
    --source-type cluster
```

Sortie :

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "cluster",
            "Events": [
                {
                    "EventId": "REDSHIFT-EVENT-2000",
                    "EventCategories": [
```

```
                        "management"
                    ],
                    "EventDescription": "Cluster <cluster name> created at <time in
 UTC>.",
                    "Severity": "INFO"
                },
                {
                    "EventId": "REDSHIFT-EVENT-2001",
                    "EventCategories": [
                        "management"
                    ],
                    "EventDescription": "Cluster <cluster name> deleted at <time in
 UTC>.",
                    "Severity": "INFO"
                },
                {
                    "EventId": "REDSHIFT-EVENT-3625",
                    "EventCategories": [
                        "monitoring"
                    ],
                    "EventDescription": "The cluster <cluster name> can't be resumed
 with its previous elastic network interface <ENI id>. We will allocate a new
 elastic network interface and associate it with the cluster node.",
                    "Severity": "INFO"
                }
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeEventCategoriesla section Référence des AWS CLI commandes.

`describe-event-subscriptions`

L'exemple de code suivant montre comment utiliser`describe-event-subscriptions`.

AWS CLI

Pour décrire les abonnements aux événements

L'`describe-event-subscriptions`exemple suivant affiche les abonnements aux notifications d'événements pour l'abonnement spécifié.

```
aws redshift describe-event-subscriptions \
    --subscription-name mysubscription
```

Sortie :

```
{
    "EventSubscriptionsList": [
        {
            "CustomerAwsId": "123456789012",
            "CustSubscriptionId": "mysubscription",
            "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
            "Status": "active",
            "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",
            "SourceIdsList": [],
            "EventCategoriesList": [
                "management"
            ],
            "Severity": "ERROR",
            "Enabled": true,
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Abonnement aux notifications d'événements Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DescribeEventSubscriptionsla section Référence des AWS CLI commandes.

`describe-events`

L'exemple de code suivant montre comment utiliser`describe-events`.

AWS CLI

Décrire tous les événements Cet exemple renvoie tous les événements. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-events
```

Résultat:

```
{
   "Events": [
      {
      "Date": "2013-01-22T19:17:03.640Z",
      "SourceIdentifier": "myclusterparametergroup",
      "Message": "Cluster parameter group myclusterparametergroup has been
 created.",
      "SourceType": "cluster-parameter-group"
      } ],
   "ResponseMetadata": {
      "RequestId": "9f056111-64c9-11e2-9390-ff04f2c1e638"
   }
}
```

Vous pouvez également obtenir les mêmes informations au format texte à l'aide de l'`--output text`option .Commande :

`--output text`Option. Commande :

Option. Commande :

```
aws redshift describe-events --output text
```

Résultat:

```
2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
 myclusterparametergroup has been created.       cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17
```

- Pour plus de détails sur l'API, voir DescribeEventsla section Référence des AWS CLI commandes.

`describe-hsm-client-certificates`

L'exemple de code suivant montre comment utiliser`describe-hsm-client-certificates`.

AWS CLI

Pour décrire les certificats clients HSM

L'`describe-hsm-client-certificates`exemple suivant affiche les détails du certificat client HSM spécifié.

```
aws redshift describe-hsm-client-certificates \
    --hsm-client-certificate-identifier myhsmclientcert
```

Sortie :

```
{
    "HsmClientCertificates": [
        {
        "HsmClientCertificateIdentifier": "myhsmclientcert",
        "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\
        EXAMPLECAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
        VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
        b24xFDASBgNVBAsTC0lBTSBDb25zEXAMPLEwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
        BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN
        EXAMPLE0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
        VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
        b2xlMRIwEAYDVQQDEwlUZXN0Q2lsEXAMPLEdBgkqhkiG9w0BCQEWEG5vb25lQGFt
        YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEXAMPLEMaK0dn+a4GmWIWJ
        21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
        rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY67EXAMPLEE
        EXAMPLEZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
        nUhVVxYUntneD9EXAMPLE6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
        FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
        NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
        "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez la référence des autorisations d'API Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DescribeHsmClientCertificatesla section Référence des AWS CLI commandes.

`describe-hsm-configurations`

L'exemple de code suivant montre comment utiliser`describe-hsm-configurations`.

AWS CLI

Pour décrire les configurations HSM

L'`describe-hsm-configurations`exemple suivant affiche les détails des configurations HSM disponibles pour le AWS compte appelant.

```
aws redshift describe-hsm-configurations /
    --hsm-configuration-identifier myhsmconnection
```

Sortie :

```
{
    "HsmConfigurations": [
        {
            "HsmConfigurationIdentifier": "myhsmconnection",
            "Description": "My HSM connection",
            "HsmIpAddress": "192.0.2.09",
            "HsmPartitionName": "myhsmpartition",
            "Tags": []
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeHsmConfigurationsla section Référence des AWS CLI commandes.

`describe-logging-status`

L'exemple de code suivant montre comment utiliser`describe-logging-status`.

AWS CLI

Pour décrire l'état de journalisation d'un cluster

L'`describe-logging-status`exemple suivant indique si des informations, telles que les requêtes et les tentatives de connexion, sont enregistrées pour un cluster.

```
aws redshift describe-logging-status \
    --cluster-identifier mycluster
```

Sortie :

```
{
    "LoggingEnabled": false
}
```

Pour plus d'informations, consultez la section Journalisation des audits de base de données dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DescribeLoggingStatusla section Référence des AWS CLI commandes.

`describe-node-configuration-options`

L'exemple de code suivant montre comment utiliser`describe-node-configuration-options`.

AWS CLI

Pour décrire les options de configuration des nœuds

L'`describe-node-configuration-options`exemple suivant affiche les propriétés des configurations de nœuds possibles, telles que le type de nœud, le nombre de nœuds et l'utilisation du disque pour le snapshot de cluster spécifié.

```
aws redshift describe-node-configuration-options \
    --action-type restore-cluster \
    --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

Sortie :

```
{
    "NodeConfigurationOptionList": [
        {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "EstimatedDiskUtilizationPercent": 19.61
        },
        {
            "NodeType": "dc2.large",
            "NumberOfNodes": 4,
            "EstimatedDiskUtilizationPercent": 9.96
```

```
        },
        {
            "NodeType": "ds2.xlarge",
            "NumberOfNodes": 2,
            "EstimatedDiskUtilizationPercent": 1.53
        },
        {
            "NodeType": "ds2.xlarge",
            "NumberOfNodes": 4,
            "EstimatedDiskUtilizationPercent": 0.78
        }
    ]
}
```

Pour plus d'informations, consultez la section Achat de nœuds réservés Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, voir DescribeNodeConfigurationOptionsla section Référence des AWS CLI commandes.

`describe-orderable-cluster-options`

L'exemple de code suivant montre comment utiliser`describe-orderable-cluster-options`.

AWS CLI

L' OptionsThis exemple Décrivant tous les clusters commandables renvoie des descriptions de toutes les options de cluster commandables. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift describe-orderable-cluster-options
```

Résultat:

```
{
   "OrderableClusterOptions": [
      {
         "NodeType": "dw.hs1.8xlarge",
         "AvailabilityZones": [
            { "Name": "us-east-1a" },
            { "Name": "us-east-1b" },
            { "Name": "us-east-1c" } ],
```

```
         "ClusterVersion": "1.0",
         "ClusterType": "multi-node"
      },
      {
         "NodeType": "dw.hs1.xlarge",
         "AvailabilityZones": [
            { "Name": "us-east-1a" },
            { "Name": "us-east-1b" },
            { "Name": "us-east-1c" } ],
         "ClusterVersion": "1.0",
         "ClusterType": "multi-node"
      },
      {
      "NodeType": "dw.hs1.xlarge",
      "AvailabilityZones": [
         { "Name": "us-east-1a" },
         { "Name": "us-east-1b" },
         { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "single-node"
      } ],
   "ResponseMetadata": {
      "RequestId": "f6000035-64cb-11e2-9135-ff82df53a51a"
   }
}
```

Vous pouvez également obtenir les mêmes informations au format texte à l'aide de l'`--output text`option .Commande :

`--output text`Option. Commande :

Option. Commande :

```
aws redshift describe-orderable-cluster-options --output text
```

Résultat:

```
dw.hs1.8xlarge      1.0     multi-node
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge       1.0     multi-node
```

```
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge       1.0     single-node
us-east-1a
us-east-1b
us-east-1c
RESPONSEMETADATA    e648696b-64cb-11e2-bec0-17624ad140dd
```

- Pour plus de détails sur l'API, reportez-vous DescribeOrderableClusterOptionsà la section Référence des AWS CLI commandes.

`describe-reserved-node-offerings`

L'exemple de code suivant montre comment utiliser`describe-reserved-node-offerings`.

AWS CLI

L' OfferingsThis exemple de description du nœud réservé montre toutes les offres de nœuds réservés disponibles à l'achat.Commande :

```
aws redshift describe-reserved-node-offerings
```

Résultat:

```
{
   "ReservedNodeOfferings": [
      {
         "OfferingType": "Heavy Utilization",
         "FixedPrice": "",
         "NodeType": "dw.hs1.xlarge",
         "UsagePrice": "",
         "RecurringCharges": [
            {
               "RecurringChargeAmount": "",
               "RecurringChargeFrequency": "Hourly"
            } ],
         "Duration": 31536000,
         "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
      },
      {
         "OfferingType": "Heavy Utilization",
```

```
            "FixedPrice": "",
            "NodeType": "dw.hs1.8xlarge",
            "UsagePrice": "",
            "RecurringCharges": [
               {
               "RecurringChargeAmount": "",
               "RecurringChargeFrequency": "Hourly"
               } ],
            "Duration": 31536000,
            "ReservedNodeOfferingId": "e5a2ff3b-352d-4a9c-ad7d-373c4cab5dd2"
         },
         ...remaining output omitted...
      ],
      "ResponseMetadata": {
         "RequestId": "8b1a1a43-75ff-11e2-9666-e142fe91ddd1"
      }
   }
```

Si vous souhaitez acheter une offre de nœuds réservés, vous pouvez appeler `purchase-reserved-node-offering` en utilisant un numéro valide ReservedNodeOfferingId.

`purchase-reserved-node-offering`en utilisant un valide ReservedNodeOfferingId.

en utilisant un valide ReservedNodeOfferingId.

ReservedNodeOfferingId.

.

- Pour plus de détails sur l'API, reportez-vous DescribeReservedNodeOfferingsà la section Référence des AWS CLI commandes.

`describe-reserved-nodes`

L'exemple de code suivant montre comment utiliser`describe-reserved-nodes`.

AWS CLI

L' NodesThis exemple de description des nœuds réservés montre une offre de nœuds réservés qui a été achetée.Commande :

```
aws redshift describe-reserved-nodes
```

Résultat:

```
{
   "ResponseMetadata": {
      "RequestId": "bc29ce2e-7600-11e2-9949-4b361e7420b7"
   },
   "ReservedNodes": [
      {
         "OfferingType": "Heavy Utilization",
         "FixedPrice": "",
         "NodeType": "dw.hs1.xlarge",
         "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
         "UsagePrice": "",
         "RecurringCharges": [
            {
               "RecurringChargeAmount": "",
               "RecurringChargeFrequency": "Hourly"
            } ],
         "NodeCount": 1,
         "State": "payment-pending",
         "StartTime": "2013-02-13T17:08:39.051Z",
         "Duration": 31536000,
         "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
      }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeReservedNodesà la section Référence des AWS CLI commandes.

`describe-resize`

L'exemple de code suivant montre comment utiliser`describe-resize`.

AWS CLI

L' ResizeThis exemple de description décrit le dernier redimensionnement d'un cluster. La demande portait sur 3 nœuds de type `dw.hs1.8xlarge` .Command :

```
aws redshift describe-resize --cluster-identifier mycluster
```

Résultat:

```
{
   "Status": "NONE",
   "TargetClusterType": "multi-node",
   "TargetNodeType": "dw.hs1.8xlarge",
   "ResponseMetadata": {
      "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"
   },
   "TargetNumberOfNodes": "3"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeResizeà la section Référence des AWS CLI commandes.

`describe-scheduled-actions`

L'exemple de code suivant montre comment utiliser`describe-scheduled-actions`.

AWS CLI

Pour décrire les actions planifiées

L'`describe-scheduled-actions`exemple suivant affiche les détails de toutes les actions actuellement planifiées.

```
aws redshift describe-scheduled-actions
```

Sortie :

```
{
    "ScheduledActions": [
        {
            "ScheduledActionName": "resizecluster",
            "TargetAction": {
                "ResizeCluster": {
                    "ClusterIdentifier": "mycluster",
                    "NumberOfNodes": 4,
                    "Classic": false
                }
            },
            "Schedule": "at(2019-12-10T00:07:00)",
            "IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",
            "State": "ACTIVE",
```

```
            "NextInvocations": [
                "2019-12-10T00:07:00Z"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeScheduledActionsà la section Référence des AWS CLI commandes.

`describe-snapshot-copy-grants`

L'exemple de code suivant montre comment utiliser`describe-snapshot-copy-grants`.

AWS CLI

Pour décrire les autorisations de copie instantanée

L'`describe-snapshot-copy-grants`exemple suivant affiche les détails de l'autorisation de copie instantanée du cluster spécifiée.

```
aws redshift describe-snapshot-copy-grants \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

Sortie :

```
{
    "SnapshotCopyGrants": [
        {
            "SnapshotCopyGrantName": "mysnapshotcopygrantname",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez Amazon Redshift Database Encryption dans le manuel Amazon Redshift Cluster Management Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeSnapshotCopyGrantsà la section Référence des AWS CLI commandes.

`describe-snapshot-schedules`

L'exemple de code suivant montre comment utiliser`describe-snapshot-schedules`.

AWS CLI

Pour décrire les plannings de snapshots

L'`describe-snapshot-schedules`exemple suivant affiche les détails du calendrier de capture d'écran du cluster spécifié.

```
aws redshift describe-snapshot-schedules \
    --cluster-identifier mycluster \
    --schedule-identifier mysnapshotschedule
```

Sortie :

```
{
    "SnapshotSchedules": [
        {
            "ScheduleDefinitions": [
                "rate(12 hours)"
            ],
            "ScheduleIdentifier": "mysnapshotschedule",
            "ScheduleDescription": "My schedule description",
            "Tags": [],
            "AssociatedClusterCount": 1,
            "AssociatedClusters": [
                {
                    "ClusterIdentifier": "mycluster",
                    "ScheduleAssociationState": "ACTIVE"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Planification automatisée des instantanés dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous DescribeSnapshotSchedulesà la section Référence des AWS CLI commandes.

`describe-storage`

L'exemple de code suivant montre comment utiliser`describe-storage`.

AWS CLI

Pour décrire le stockage

L'`describe-storage`exemple suivant affiche des détails sur le stockage de sauvegarde et les tailles de stockage provisoires du compte.

```
aws redshift describe-storage
```

Sortie :

```
{
    "TotalBackupSizeInMegaBytes": 193149.0,
    "TotalProvisionedStorageInMegaBytes": 655360.0
}
```

Pour plus d'informations, consultez la section Gestion du stockage des snapshots dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous DescribeStorageà la section Référence des AWS CLI commandes.

`describe-table-restore-status`

L'exemple de code suivant montre comment utiliser`describe-table-restore-status`.

AWS CLI

Pour décrire l'état des demandes de restauration de tables à partir d'un instantané de cluster

L'`describe-table-restore-status`exemple suivant affiche les détails des demandes de restauration de table effectuées pour le cluster spécifié.

```
aws redshift describe-table-restore-status /
    --cluster-identifier mycluster
```

Sortie :

```
{
    "TableRestoreStatusDetails": [
        {
            "TableRestoreRequestId": "z1116630-0e80-46f4-ba86-bd9670411ebd",
            "Status": "IN_PROGRESS",
            "RequestTime": "2019-12-27T18:22:12.257Z",
            "ClusterIdentifier": "mycluster",
            "SnapshotIdentifier": "mysnapshotid",
            "SourceDatabaseName": "dev",
            "SourceSchemaName": "public",
            "SourceTableName": "mytable",
            "TargetDatabaseName": "dev",
            "TargetSchemaName": "public",
            "NewTableName": "mytable-clone"
        }
    ]
}
```

Pour plus d'informations, consultez Restaurer une table à partir d'un instantané dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous DescribeTableRestoreStatusà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire les balises

L'`describe-tags`exemple suivant affiche les ressources que le cluster spécifié a associées aux noms et valeurs de balises spécifiés.

```
aws redshift describe-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tag-keys clustertagkey \
    --tag-values clustertagvalue
```

Sortie :

```
{
    "TaggedResources": [
                    {
            "Tag": {
                "Key": "clustertagkey",
                "Value": "clustertagvalue"
            },
            "ResourceName": "arn:aws:redshift:us-
west-2:123456789012:cluster:mycluster",
            "ResourceType": "cluster"
        }
    ]
}
```

Pour plus d'informations, consultez la section Ressources de balisage dans Amazon Redshift dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous DescribeTagsà la section Référence des AWS CLI commandes.

`disable-snapshot-copy`

L'exemple de code suivant montre comment utiliser`disable-snapshot-copy`.

AWS CLI

Pour désactiver la copie instantanée d'un cluster

L'`disable-snapshot-copy`exemple suivant désactive la copie automatique d'un instantané pour le cluster spécifié.

```
aws redshift disable-snapshot-copy \
    --cluster-identifier mycluster
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
```

```
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-i9b431cd",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1fel7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [
```

```
            {
                "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
                "ApplyStatus": "in-sync"
            }
        ],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

Pour plus d'informations, consultez Copier des instantanés vers une autre AWS région dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous DisableSnapshotCopyà la section Référence des AWS CLI commandes.

`enable-snapshot-copy`

L'exemple de code suivant montre comment utiliser`enable-snapshot-copy`.

AWS CLI

Pour activer la copie instantanée d'un cluster

L'`enable-snapshot-copy`exemple suivant active la copie automatique d'un instantané pour le cluster spécifié.

```
aws redshift enable-snapshot-copy \
    --cluster-identifier mycluster \
    --destination-region us-west-1
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
```

```
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-f4c731cd",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1ael7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 7,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
```

```
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
                "ApplyStatus": "in-sync"
            }
        ],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

Pour plus d'informations, consultez Copier des instantanés vers une autre AWS région dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous EnableSnapshotCopyà la section Référence des AWS CLI commandes.

`get-cluster-credentials`

L'exemple de code suivant montre comment utiliser`get-cluster-credentials`.

AWS CLI

Pour obtenir les informations d'identification du cluster pour un AWS compte

L'`get-cluster-credentials`exemple suivant récupère les informations d'identification temporaires qui permettent d'accéder à une base de données Amazon Redshift.

```
aws redshift get-cluster-credentials \
    --db-user adminuser --db-name dev \
    --cluster-identifier mycluster
```

Sortie :

```
{
    "DbUser": "IAM:adminuser",
```

```
    "DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx1O9d6UmpkBtvEeqFly/
EXAMPLE==",
    "Expiration": "2019-12-10T17:25:05.770Z"
}
```

Pour plus d'informations, consultez la section Génération d'informations d'identification de base de données IAM à l'aide de la CLI ou de l'API Amazon Redshift dans le guide de gestion du cluster Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous GetClusterCredentialsà la section Référence des AWS CLI commandes.

`get-reserved-node-exchange-offerings`

L'exemple de code suivant montre comment utiliser`get-reserved-node-exchange-offerings`.

AWS CLI

Pour obtenir des offres d'échange de nœuds réservés

L'`get-reserved-node-exchange-offerings`exemple suivant extrait un tableau DC2 `ReservedNodeOfferings` correspondant au nœud DC1 réservé spécifié.

```
aws redshift get-reserved-node-exchange-offerings \
    --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

Sortie :

```
{
    "ReservedNodeOfferings": [
        {
            "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
            "NodeType": "dc2.large",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CurrencyCode": "USD",
            "OfferingType": "All Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.0,
                    "RecurringChargeFrequency": "Hourly"
```

```
                }
            ],
            "ReservedNodeOfferingType": "Regular"
        }
    ]
}
```

Pour plus d'informations, consultez la section Mise à niveau des nœuds réservés avec la AWS CLI dans le manuel Amazon Redshift Cluster Management Guide.

- Pour plus de détails sur l'API, reportez-vous GetReservedNodeExchangeOfferingsà la section Référence des AWS CLI commandes.

`modify-cluster-iam-roles`

L'exemple de code suivant montre comment utiliser`modify-cluster-iam-roles`.

AWS CLI

Pour modifier le rôle IAM d'un cluster

L'`modify-cluster-iam-roles`exemple suivant supprime le rôle AWS IAM spécifié du cluster spécifié.

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
```

```
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-f9b731sd",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b2fal7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 7,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
```

```
        "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

Pour plus d'informations, consultez la section Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifyClusterIamRolesà la section Référence des AWS CLI commandes.

`modify-cluster-maintenance`

L'exemple de code suivant montre comment utiliser`modify-cluster-maintenance`.

AWS CLI

Pour modifier la maintenance du cluster

L'`modify-cluster-maintenance`exemple suivant reporte de 30 jours la maintenance du cluster spécifié.

```
aws redshift modify-cluster-maintenance \
    --cluster-identifier mycluster \
    --defer-maintenance \
    --defer-maintenance-duration 30
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
```

```
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1ael7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 7,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [
```

```
            {
                "DeferMaintenanceIdentifier": "dfm-mUdVIfFcT1B4SGhw6fyF",
                "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
                "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
            }
        ],
        "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
    }
}
```

Pour plus d'informations, consultez la section Maintenance des clusters dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifyClusterMaintenanceà la section Référence des AWS CLI commandes.

`modify-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`modify-cluster-parameter-group`.

AWS CLI

Modifier un paramètre dans un groupe de paramètres

L'`modify-cluster-parameter-group`exemple suivant modifie le paramètre wlm_json_configuration pour la gestion de la charge de travail. Il accepte les paramètres d'un fichier contenant le contenu JSON indiqué ci-dessous.

```
aws redshift modify-cluster-parameter-group \
    --parameter-group-name myclusterparametergroup \
    --parameters file://modify_pg.json
```

Contenu de `modify_pg.json` :

```
[
    {
        "ParameterName": "wlm_json_configuration",
        "ParameterValue": "[{\"user_group\":\"example_user_group1\",\"query_group\":
 \"example_query_group1\", \"query_concurrency\":7},{\"query_concurrency\":5}]"
```

```
    }
]
```

Sortie :

```
{
   "ParameterGroupStatus": "Your parameter group has been updated but changes won't
 get applied until you reboot the associated Clusters.",
   "ParameterGroupName": "myclusterparametergroup",
   "ResponseMetadata": {
      "RequestId": "09974cc0-64cd-11e2-bea9-49e0ce183f07"
   }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyClusterParameterGroupà la section Référence des AWS CLI commandes.

`modify-cluster-snapshot-schedule`

L'exemple de code suivant montre comment utiliser`modify-cluster-snapshot-schedule`.

AWS CLI

Pour modifier le calendrier des instantanés du cluster

L'`modify-cluster-snapshot-schedule`exemple suivant supprime le calendrier de capture spécifié du cluster spécifié.

```
aws redshift modify-cluster-snapshot-schedule \
    --cluster-identifier mycluster \
    --schedule-identifier mysnapshotschedule \
    --disassociate-schedule
```

Cette commande ne génère pas de sortie.

Pour plus d'informations, consultez la section Planification automatisée des instantanés dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifyClusterSnapshotScheduleà la section Référence des AWS CLI commandes.

`modify-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`modify-cluster-snapshot`.

AWS CLI

Pour modifier un instantané du cluster

L'`modify-cluster-snapshot`exemple suivant définit le paramètre de période de rétention manuelle pour le cliché de cluster spécifié sur une valeur de 10 jours.

```
aws redshift modify-cluster-snapshot \
    --snapshot-identifier mycluster-2019-11-06-16-32 \
    --manual-snapshot-retention-period 10
```

Sortie :

```
{
    "Snapshot": {
        "SnapshotIdentifier": "mycluster-2019-11-06-16-32",
        "ClusterIdentifier": "mycluster",
        "SnapshotCreateTime": "2019-12-07T00:34:05.633Z",
        "Status": "available",
        "Port": 5439,
        "AvailabilityZone": "us-west-2f",
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "MasterUsername": "adminuser",
        "ClusterVersion": "1.0",
        "SnapshotType": "manual",
        "NodeType": "dc2.large",
        "NumberOfNodes": 2,
        "DBName": "dev",
        "VpcId": "vpc-b1cel7t9",
        "Encrypted": false,
        "EncryptedWithHSM": false,
        "OwnerAccount": "123456789012",
        "TotalBackupSizeInMegaBytes": 64384.0,
        "ActualIncrementalBackupSizeInMegaBytes": 24.0,
        "BackupProgressInMegaBytes": 24.0,
        "CurrentBackupRateInMegaBytesPerSecond": 13.0011,
        "EstimatedSecondsToCompletion": 0,
        "ElapsedTimeInSeconds": 1,
        "Tags": [
```

```
            {
                "Key": "mytagkey",
                "Value": "mytagvalue"
            }
        ],
        "EnhancedVpcRouting": false,
        "MaintenanceTrackName": "current",
        "ManualSnapshotRetentionPeriod": 10,
        "ManualSnapshotRemainingDays": 6,
        "SnapshotRetentionStartTime": "2019-12-07T00:34:07.479Z"
    }
}
```

Pour plus d'informations, consultez Amazon Redshift Snapshots dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifyClusterSnapshotà la section Référence des AWS CLI commandes.

`modify-cluster-subnet-group`

L'exemple de code suivant montre comment utilisermodify-cluster-subnet-group.

AWS CLI

Modifier les sous-réseaux dans un sous-réseau de cluster L' GroupThis exemple montre comment modifier la liste des sous-réseaux dans un groupe de sous-réseaux de cache. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
 --subnet-ids subnet-763fdd1 subnet-ac830e9
```

Résultat:

```
{
   "ClusterSubnetGroup":
   {
      "Subnets": [
         {
            "SubnetStatus": "Active",
            "SubnetIdentifier": "subnet-763fdd1c",
```

```
            "SubnetAvailabilityZone":
               { "Name": "us-east-1a" }
         },
         {
            "SubnetStatus": "Active",
            "SubnetIdentifier": "subnet-ac830e9",
            "SubnetAvailabilityZone":
               { "Name": "us-east-1b" }
         } ],
      "VpcId": "vpc-7e3fdd14",
      "SubnetGroupStatus": "Complete",
      "Description": "My subnet group",
      "ClusterSubnetGroupName": "mysubnetgroup"
   },
   "ResponseMetadata": {
      "RequestId": "8da93e89-8372-f936-93a8-873918938197a"
   }
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyClusterSubnetGroupà la section Référence des AWS CLI commandes.

`modify-cluster`

L'exemple de code suivant montre comment utiliser`modify-cluster`.

AWS CLI

Associer un groupe de sécurité à l'aide d'un ClusterThis exemple montre comment associer un groupe de sécurité de cluster au cluster spécifié.Commande :

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups
 mysecuritygroup
```

Modifiez la fenêtre de maintenance pour ClusterThis montrer comment modifier la fenêtre de maintenance hebdomadaire préférée pour un cluster afin qu'elle soit au minimum de quatre heures commençant le dimanche à 23 h 15 et se terminant le lundi à 3 h 15. Commande :

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-
window Sun:23:15-Mon:03:15
```

Modifiez le mot de passe principal. L' ClusterThis exemple montre comment modifier le mot de passe principal d'un cluster.Commande :

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
 A1b2c3d4
```

- Pour plus de détails sur l'API, reportez-vous ModifyClusterà la section Référence des AWS CLI commandes.

`modify-event-subscription`

L'exemple de code suivant montre comment utiliser`modify-event-subscription`.

AWS CLI

Pour modifier l'abonnement à un événement

L'`modify-event-subscription`exemple suivant désactive l'abonnement aux notifications d'événements spécifié.

```
aws redshift modify-event-subscription \
    --subscription-name mysubscription \
    --no-enabled
```

Sortie :

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "mysubscription",
        "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
        "Status": "active",
        "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",
        "SourceIdsList": [],
        "EventCategoriesList": [
            "management"
        ],
        "Severity": "ERROR",
        "Enabled": false,
        "Tags": []
    }
```

```
}
```

Pour plus d'informations, consultez la section Abonnement aux notifications d'événements Amazon Redshift dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifyEventSubscriptionà la section Référence des AWS CLI commandes.

`modify-scheduled-action`

L'exemple de code suivant montre comment utiliser`modify-scheduled-action`.

AWS CLI

Pour modifier une action planifiée

L'`modify-scheduled-action`exemple suivant ajoute une description à l'action planifiée existante spécifiée.

```
aws redshift modify-scheduled-action \
    --scheduled-action-name myscheduledaction \
    --scheduled-action-description "My scheduled action"
```

Sortie :

```
{
    "ScheduledActionName": "myscheduledaction",
    "TargetAction": {
        "ResizeCluster": {
            "ClusterIdentifier": "mycluster",
            "NumberOfNodes": 2,
            "Classic": false
        }
    },
    "Schedule": "at(2019-12-25T00:00:00)",
    "IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",
    "ScheduledActionDescription": "My scheduled action",
    "State": "ACTIVE",
    "NextInvocations": [
        "2019-12-25T00:00:00Z"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ModifyScheduledActionà la section Référence des AWS CLI commandes.

`modify-snapshot-copy-retention-period`

L'exemple de code suivant montre comment utiliser`modify-snapshot-copy-retention-period`.

AWS CLI

Pour modifier la période de conservation des copies instantanées

L'`modify-snapshot-copy-retention-period`exemple suivant modifie le nombre de jours pendant lesquels les instantanés du cluster spécifié sont conservés dans la AWS région de destination après leur copie depuis la région source AWS .

```
aws redshift modify-snapshot-copy-retention-period \
    --cluster-identifier mycluster \
    --retention-period 15
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
```

```
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1fet7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 15,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [
            {
                "DeferMaintenanceIdentifier": "dfm-mUdVSfDcT1F4SGhw6fyF",
                "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
                "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
            }
        ],
        "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
```

```
    }
}
```

Pour plus d'informations, consultez la section Snapshot Schedule Format dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifySnapshotCopyRetentionPeriodà la section Référence des AWS CLI commandes.

`modify-snapshot-schedule`

L'exemple de code suivant montre comment utiliser`modify-snapshot-schedule`.

AWS CLI

Pour modifier le calendrier des instantanés

L'`modify-snapshot-schedule`exemple suivant modifie le taux de la planification de captures d'écran spécifiée à toutes les 10 heures.

```
aws redshift modify-snapshot-schedule \
    --schedule-identifier mysnapshotschedule \
    --schedule-definitions "rate(10 hours)"
```

Sortie :

```
{
    "ScheduleDefinitions": [
        "rate(10 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
}
```

Pour plus d'informations, consultez la section Snapshot Schedule Format dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ModifySnapshotScheduleà la section Référence des AWS CLI commandes.

`purchase-reserved-node-offering`

L'exemple de code suivant montre comment utiliser`purchase-reserved-node-offering`.

AWS CLI

L' NodeThis exemple d'achat d'une offre réservée montre comment acheter une offre de nœuds réservés. `reserved-node-offering-id`Il est obtenu en appelant `describe-reserved-node-offerings` .Command :

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-cf4c-4343-be8b-d832c45ab51c
```

Résultat:

```
{
   "ReservedNode": {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.xlarge",
      "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
      "UsagePrice": "",
      "RecurringCharges": [
         {
            "RecurringChargeAmount": "",
            "RecurringChargeFrequency": "Hourly"
         }
      ],
      "NodeCount": 1,
      "State": "payment-pending",
      "StartTime": "2013-02-13T17:08:39.051Z",
      "Duration": 31536000,
      "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
   },
   "ResponseMetadata": {
      "RequestId": "01bda7bf-7600-11e2-b605-2568d7396e7f"
   }
}
```

- Pour plus de détails sur l'API, reportez-vous PurchaseReservedNodeOfferingà la section Référence des AWS CLI commandes.

`reboot-cluster`

L'exemple de code suivant montre comment utiliser`reboot-cluster`.

AWS CLI

Redémarrer un ClusterThis exemple redémarre un cluster. Par défaut, la sortie est au format JSON.Commande :

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

Résultat:

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
         "Port": 5439,
         "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
         "ParameterApplyStatus": "in-sync",
         "ParameterGroupName": "default.redshift-1.0"
         }
      ],
      "ClusterSecurityGroups": [
         {
         "Status": "active",
         "ClusterSecurityGroupName": "default"
         }
      ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "AvailabilityZone": "us-east-1a",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "rebooting",
      "ClusterIdentifier": "mycluster",
```

```
        "DBName": "dev",
        "NumberOfNodes": 2,
        "PendingModifiedValues": {}
    },
    "ResponseMetadata": {
        "RequestId": "61c8b564-64e8-11e2-8f7d-3b939af52818"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous RebootClusterà la section Référence des AWS CLI commandes.

`reset-cluster-parameter-group`

L'exemple de code suivant montre comment utiliser`reset-cluster-parameter-group`.

AWS CLI

L' GroupThis exemple de réinitialisation des paramètres dans un paramètre montre comment réinitialiser tous les paramètres d'un groupe de paramètres.Commande :

```
aws redshift reset-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --reset-all-parameters
```

- Pour plus de détails sur l'API, reportez-vous ResetClusterParameterGroupà la section Référence des AWS CLI commandes.

`resize-cluster`

L'exemple de code suivant montre comment utiliser`resize-cluster`.

AWS CLI

Pour redimensionner un cluster

L'`resize-cluster`exemple suivant redimensionne le cluster spécifié.

```
aws redshift resize-cluster \
    --cluster-identifier mycluster \
    --cluster-type multi-node \
    --node-type dc2.large \
```

```
    --number-of-nodes 6 \
    --classic
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "resizing",
        "ClusterAvailabilityStatus": "Modifying",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-a1abc1a1",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 6,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
```

```
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 15,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [
            {
                "DeferMaintenanceIdentifier": "dfm-mUdVCfDcT1B4SGhw6fyF",
                "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
                "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
            }
        ],
        "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z",
        "ResizeInfo": {
            "ResizeType": "ClassicResize",
            "AllowCancelResize": true
        }
    }
}
```

Pour plus d'informations, consultez la section Redimensionnement d'un cluster dans le guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous ResizeClusterà la section Référence des AWS CLI commandes.

`restore-from-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`restore-from-cluster-snapshot`.

AWS CLI

Restaurer un cluster à partir d'un SnapshotThis exemple restaure un cluster à partir d'un snapshot.Commande :

```
aws redshift restore-from-cluster-snapshot --cluster-identifier mycluster-clone --
snapshot-identifier my-snapshot-id
```

Résultat:

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
         "ParameterApplyStatus": "in-sync",
         "ParameterGroupName": "default.redshift-1.0"
         }
      ],
      "ClusterSecurityGroups": [
         {
         "Status": "active",
         "ClusterSecurityGroupName": "default"
         }
      ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "creating",
      "ClusterIdentifier": "mycluster-clone",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {}
   },
   "ResponseMetadata": {
      "RequestId": "77fd512b-64e3-11e2-8f5b-e90bd6c77476"
   }
}
```

- Pour plus de détails sur l'API, reportez-vous RestoreFromClusterSnapshotà la section Référence des AWS CLI commandes.

`restore-table-from-cluster-snapshot`

L'exemple de code suivant montre comment utiliser`restore-table-from-cluster-snapshot`.

AWS CLI

Pour restaurer une table à partir d'un instantané de cluster

L'`restore-table-from-cluster-snapshot`exemple suivant crée une nouvelle table à partir de la table spécifiée dans le cliché de cluster spécifié.

```
aws redshift restore-table-from-cluster-snapshot /
    --cluster-identifier mycluster /
    --snapshot-identifier mycluster-2019-11-19-16-17 /
    --source-database-name dev /
    --source-schema-name public /
    --source-table-name mytable /
    --target-database-name dev /
    --target-schema-name public /
    --new-table-name mytable-clone
```

Sortie :

```
{
    "TableRestoreStatus": {
        "TableRestoreRequestId": "a123a12b-abc1-1a1a-a123-a1234ab12345",
        "Status": "PENDING",
        "RequestTime": "2019-12-20T00:20:16.402Z",
        "ClusterIdentifier": "mycluster",
        "SnapshotIdentifier": "mycluster-2019-11-19-16-17",
        "SourceDatabaseName": "dev",
        "SourceSchemaName": "public",
        "SourceTableName": "mytable",
        "TargetDatabaseName": "dev",
        "TargetSchemaName": "public",
        "NewTableName": "mytable-clone"
    }
}
```

Pour plus d'informations, consultez Restaurer une table à partir d'un instantané dans le Guide de gestion des clusters Amazon Redshift.

- Pour plus de détails sur l'API, reportez-vous RestoreTableFromClusterSnapshotà la section Référence des AWS CLI commandes.

`revoke-cluster-security-group-ingress`

L'exemple de code suivant montre comment utiliser`revoke-cluster-security-group-ingress`.

AWS CLI

Révoquer l'accès depuis un GroupThis exemple de EC2 sécurité révoque l'accès à un groupe de EC2 sécurité Amazon nommé.Commande :

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

Révocation de l'accès à une plage CIDRCet exemple révoque l'accès à une plage CIDR.Commande :

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- Pour plus de détails sur l'API, reportez-vous RevokeClusterSecurityGroupIngressà la section Référence des AWS CLI commandes.

`revoke-snapshot-access`

L'exemple de code suivant montre comment utiliser`revoke-snapshot-access`.

AWS CLI

Révoquer l'autorisation d'un AWS compte pour restaurer un SnapshotThis exemple révoque l'autorisation du AWS compte `444455556666` pour restaurer l'instantané. `my-snapshot-id` Par défaut, la sortie est au format JSON.Commande :

```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

Résultat:

```
{
   "Snapshot": {
      "Status": "available",
      "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
      "EstimatedSecondsToCompletion": 0,
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "Encrypted": false,
      "OwnerAccount": "111122223333",
      "BackupProgressInMegabytes": 11.0,
      "ElapsedTimeInSeconds": 0,
      "DBName": "dev",
      "CurrentBackupRateInMegabytesPerSecond: 0.1534,
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "ActualIncrementalBackupSizeInMegabytes"; 11.0,
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "TotalBackupSizeInMegabytes": 20.0,
      "Port": 5439,
      "NumberOfNodes": 2,
      "SnapshotIdentifier": "my-snapshot-id"
   }
}
```

- Pour plus de détails sur l'API, reportez-vous RevokeSnapshotAccessà la section Référence des AWS CLI commandes.

`rotate-encryption-key`

L'exemple de code suivant montre comment utiliser`rotate-encryption-key`.

AWS CLI

Pour faire pivoter la clé de chiffrement d'un cluster

L'`rotate-encryption-key`exemple suivant fait pivoter la clé de chiffrement pour le cluster spécifié.

```
aws redshift rotate-encryption-key \
    --cluster-identifier mycluster
```

Sortie :

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "rotating-keys",
        "ClusterAvailabilityStatus": "Modifying",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-10T19:25:45.886Z",
        "AutomatedSnapshotRetentionPeriod": 30,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-a1abc1a1",
        "AvailabilityZone": "us-west-2a",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {},
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
```

```
        "NumberOfNodes": 2,
        "PubliclyAccessible": false,
        "Encrypted": true,
        "Tags": [],
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
        "EnhancedVpcRouting": false,
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
                "ApplyStatus": "in-sync"
            }
        ],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

Pour plus d'informations, consultez Amazon Redshift Database Encryption dans le manuel Amazon Redshift Cluster Management Guide.

- Pour plus de détails sur l'API, reportez-vous RotateEncryptionKeyà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon Rekognition AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Rekognition.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`compare-faces`

L'exemple de code suivant montre comment utiliser`compare-faces`.

Pour de plus amples informations, veuillez consulter Comparaison de visages dans des images.

AWS CLI

Pour comparer des visages sur deux images

La `compare-faces` commande suivante compare les visages de deux images stockées dans un compartiment Amazon S3.

```
aws rekognition compare-faces \
    --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \
    --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

Sortie :

```
{
    "UnmatchedFaces": [],
    "FaceMatches": [
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.12368916720151901,
                    "Top": 0.16007372736930847,
                    "Left": 0.5901257991790771,
                    "Height": 0.25140416622161865
                },
                "Confidence": 100.0,
                "Pose": {
                    "Yaw": -3.7351467609405518,
                    "Roll": -0.10309021919965744,
                    "Pitch": 0.8637830018997192
                },
                "Quality": {
                    "Sharpness": 95.51618957519531,
                    "Brightness": 65.29893493652344
                },
                "Landmarks": [
```

```
                    {
                        "Y": 0.26721030473709106,
                        "X": 0.6204193830490112,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.26831310987472534,
                        "X": 0.6776827573776245,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.3514654338359833,
                        "X": 0.6241428852081299,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.35258132219314575,
                        "X": 0.6713621020317078,
                        "Type": "mouthRight"
                    },
                    {
                        "Y": 0.3140771687030792,
                        "X": 0.6428444981575012,
                        "Type": "nose"
                    }
                ]
            },
            "Similarity": 100.0
        }
    ],
    "SourceImageFace": {
        "BoundingBox": {
            "Width": 0.12368916720151901,
            "Top": 0.16007372736930847,
            "Left": 0.5901257991790771,
            "Height": 0.25140416622161865
        },
        "Confidence": 100.0
    }
}
```

Pour plus d'informations, consultez la section Comparaison de visages dans des images dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CompareFacesà la section Référence des AWS CLI commandes.

`create-collection`

L'exemple de code suivant montre comment utiliser`create-collection`.

Pour plus d'informations, consultez Création d'une collection.

AWS CLI

Pour créer une collection

La `create-collection` commande suivante crée une collection portant le nom spécifié.

```
aws rekognition create-collection \
    --collection-id "MyCollection"
```

Sortie :

```
{
    "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
    "FaceModelVersion": "4.0",
    "StatusCode": 200
}
```

Pour plus d'informations, consultez la section Création d'une collection dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateCollectionà la section Référence des AWS CLI commandes.

`create-stream-processor`

L'exemple de code suivant montre comment utiliser`create-stream-processor`.

AWS CLI

Pour créer un nouveau processeur de flux

L'`create-stream-processor`exemple suivant crée un nouveau processeur de flux avec la configuration spécifiée.

```
aws rekognition create-stream-processor --name my-stream-processor\
    --input '{"KinesisVideoStream":{"Arn":"arn:aws:kinesisvideo:us-
west-2:123456789012:stream/macwebcam/1530559711205"}}'\
    --stream-processor-output '{"KinesisDataStream":{"Arn":"arn:aws:kinesis:us-
west-2:123456789012:stream/AmazonRekognitionRekStream"}}'\
    --role-arn arn:aws:iam::123456789012:role/AmazonRekognitionDetect\
    --settings '{"FaceSearch":
{"CollectionId":"MyCollection","FaceMatchThreshold":85.5}}'
```

Sortie :

```
{
    "StreamProcessorArn": "arn:aws:rekognition:us-
west-2:123456789012:streamprocessor/my-stream-processor"
}
```

Pour plus d'informations, consultez la section Utilisation de vidéos en streaming dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateStreamProcessorà la section Référence des AWS CLI commandes.

`delete-collection`

L'exemple de code suivant montre comment utiliser`delete-collection`.

Pour plus d'informations, consultez Suppression d'une collection.

AWS CLI

Pour supprimer une collection

La `delete-collection` commande suivante supprime la collection spécifiée.

```
aws rekognition delete-collection \
    --collection-id MyCollection
```

Sortie :

```
{
    "StatusCode": 200
}
```

Pour plus d'informations, consultez Supprimer une collection dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteCollectionà la section Référence des AWS CLI commandes.

`delete-faces`

L'exemple de code suivant montre comment utiliser`delete-faces`.

Pour plus d'informations, veuillez consulter Supprimer des visages d'une collection.

AWS CLI

Pour supprimer des visages d'une collection

La `delete-faces` commande suivante supprime le visage spécifié d'une collection.

```
aws rekognition delete-faces \
    --collection-id MyCollection
    --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

Sortie :

```
{
    "DeletedFaces": [
        "0040279c-0178-436e-b70a-e61b074e96b0"
    ]
}
```

Pour plus d'informations, consultez Supprimer des visages d'une collection dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteFacesà la section Référence des AWS CLI commandes.

`delete-stream-processor`

L'exemple de code suivant montre comment utiliser`delete-stream-processor`.

AWS CLI

Pour supprimer un processeur de flux

La `delete-stream-processor` commande suivante supprime le processeur de flux spécifié.

```
aws rekognition delete-stream-processor \
    --name my-stream-processor
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de vidéos en streaming dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteStreamProcessorà la section Référence des AWS CLI commandes.

`describe-collection`

L'exemple de code suivant montre comment utiliser`describe-collection`.

Pour plus d'informations, veuillez consulter Description d'une collection.

AWS CLI

Pour décrire une collection

L'`describe-collection`exemple suivant affiche les détails de la collection spécifiée.

```
aws rekognition describe-collection \
    --collection-id MyCollection
```

Sortie :

```
{
    "FaceCount": 200,
    "CreationTimestamp": 1569444828.274,
    "CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
```

```
    "FaceModelVersion": "4.0"
}
```

Pour plus d'informations, consultez la section Décrire une collection dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeCollectionà la section Référence des AWS CLI commandes.

`describe-stream-processor`

L'exemple de code suivant montre comment utiliser`describe-stream-processor`.

AWS CLI

Pour obtenir des informations sur un processeur de flux

La `describe-stream-processor` commande suivante affiche des informations détaillées sur le processeur de flux spécifié.

```
aws rekognition describe-stream-processor \
    --name my-stream-processor
```

Sortie :

```
{
    "Status": "STOPPED",
    "Name": "my-stream-processor",
    "LastUpdateTimestamp": 1532449292.712,
    "Settings": {
        "FaceSearch": {
            "FaceMatchThreshold": 80.0,
            "CollectionId": "my-collection"
        }
    },
    "RoleArn": "arn:aws:iam::123456789012:role/AmazonRekognitionDetectStream",
    "StreamProcessorArn": "arn:aws:rekognition:us-
west-2:123456789012:streamprocessor/my-stream-processpr",
    "Output": {
        "KinesisDataStream": {
            "Arn": "arn:aws:kinesis:us-west-2:123456789012:stream/
AmazonRekognitionRekStream"
        }
```

```
    },
    "Input": {
        "KinesisVideoStream": {
            "Arn": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
macwebcam/123456789012"
        }
    },
    "CreationTimestamp": 1532449292.712
}
```

Pour plus d'informations, consultez la section Utilisation de vidéos en streaming dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeStreamProcessorà la section Référence des AWS CLI commandes.

`detect-faces`

L'exemple de code suivant montre comment utiliser`detect-faces`.

Pour plus d'informations, veuillez consulter Détecter des visages dans une image.

AWS CLI

Pour détecter des visages dans une image

La `detect-faces` commande suivante détecte les visages dans l'image spécifiée stockée dans un compartiment Amazon S3.

```
aws rekognition detect-faces \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \
    --attributes "ALL"
```

Sortie :

```
{
    "FaceDetails": [
        {
            "Confidence": 100.0,
            "Eyeglasses": {
                "Confidence": 98.91107940673828,
                "Value": false
            },
```

```
            "Sunglasses": {
                "Confidence": 99.7966537475586,
                "Value": false
            },
            "Gender": {
                "Confidence": 99.56611633300781,
                "Value": "Male"
            },
            "Landmarks": [
                {
                    "Y": 0.26721030473709106,
                    "X": 0.6204193830490112,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.26831310987472534,
                    "X": 0.6776827573776245,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.3514654338359833,
                    "X": 0.6241428852081299,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.35258132219314575,
                    "X": 0.6713621020317078,
                    "Type": "mouthRight"
                },
                {
                    "Y": 0.3140771687030792,
                    "X": 0.6428444981575012,
                    "Type": "nose"
                },
                {
                    "Y": 0.24662546813488007,
                    "X": 0.6001564860343933,
                    "Type": "leftEyeBrowLeft"
                },
                {
                    "Y": 0.24326619505882263,
                    "X": 0.6303644776344299,
                    "Type": "leftEyeBrowRight"
                },
```

```
                {
                    "Y": 0.23818562924861908,
                    "X": 0.6146903038024902,
                    "Type": "leftEyeBrowUp"
                },
                {
                    "Y": 0.24373626708984375,
                    "X": 0.6640064716339111,
                    "Type": "rightEyeBrowLeft"
                },
                {
                    "Y": 0.24877218902111053,
                    "X": 0.7025929093360901,
                    "Type": "rightEyeBrowRight"
                },
                {
                    "Y": 0.23938551545143127,
                    "X": 0.6823262572288513,
                    "Type": "rightEyeBrowUp"
                },
                {
                    "Y": 0.265746533870697,
                    "X": 0.6112898588180542,
                    "Type": "leftEyeLeft"
                },
                {
                    "Y": 0.2676128149032593,
                    "X": 0.6317071914672852,
                    "Type": "leftEyeRight"
                },
                {
                    "Y": 0.262735515832901,
                    "X": 0.6201658248901367,
                    "Type": "leftEyeUp"
                },
                {
                    "Y": 0.27025148272514343,
                    "X": 0.6206279993057251,
                    "Type": "leftEyeDown"
                },
                {
                    "Y": 0.268223375082016,
                    "X": 0.6658390760421753,
                    "Type": "rightEyeLeft"
```

```
                },
                {
                    "Y": 0.2672517001628876,
                    "X": 0.687832236289978,
                    "Type": "rightEyeRight"
                },
                {
                    "Y": 0.26383838057518005,
                    "X": 0.6769183874130249,
                    "Type": "rightEyeUp"
                },
                {
                    "Y": 0.27138751745224,
                    "X": 0.676596462726593,
                    "Type": "rightEyeDown"
                },
                {
                    "Y": 0.32283174991607666,
                    "X": 0.6350004076957703,
                    "Type": "noseLeft"
                },
                {
                    "Y": 0.3219289481639862,
                    "X": 0.6567046642303467,
                    "Type": "noseRight"
                },
                {
                    "Y": 0.3420318365097046,
                    "X": 0.6450609564781189,
                    "Type": "mouthUp"
                },
                {
                    "Y": 0.3664324879646301,
                    "X": 0.6455618143081665,
                    "Type": "mouthDown"
                },
                {
                    "Y": 0.26721030473709106,
                    "X": 0.6204193830490112,
                    "Type": "leftPupil"
                },
                {
                    "Y": 0.26831310987472534,
                    "X": 0.6776827573776245,
```

```
                    "Type": "rightPupil"
                },
                {
                    "Y": 0.26343393325805664,
                    "X": 0.5946047306060791,
                    "Type": "upperJawlineLeft"
                },
                {
                    "Y": 0.3543180525302887,
                    "X": 0.6044883728027344,
                    "Type": "midJawlineLeft"
                },
                {
                    "Y": 0.4084877669811249,
                    "X": 0.6477024555206299,
                    "Type": "chinBottom"
                },
                {
                    "Y": 0.3562754988670349,
                    "X": 0.707981526851654,
                    "Type": "midJawlineRight"
                },
                {
                    "Y": 0.26580461859703064,
                    "X": 0.7234612107276917,
                    "Type": "upperJawlineRight"
                }
            ],
            "Pose": {
                "Yaw": -3.7351467609405518,
                "Roll": -0.10309021919965744,
                "Pitch": 0.8637830018997192
            },
            "Emotions": [
                {
                    "Confidence": 8.74203109741211,
                    "Type": "SURPRISED"
                },
                {
                    "Confidence": 2.501944065093994,
                    "Type": "ANGRY"
                },
                {
                    "Confidence": 0.7378743290901184,
```

```
                    "Type": "DISGUSTED"
                },
                {
                    "Confidence": 3.5296201705932617,
                    "Type": "HAPPY"
                },
                {
                    "Confidence": 1.7162904739379883,
                    "Type": "SAD"
                },
                {
                    "Confidence": 9.518536567687988,
                    "Type": "CONFUSED"
                },
                {
                    "Confidence": 0.45474427938461304,
                    "Type": "FEAR"
                },
                {
                    "Confidence": 72.79895782470703,
                    "Type": "CALM"
                }
            ],
            "AgeRange": {
                "High": 48,
                "Low": 32
            },
            "EyesOpen": {
                "Confidence": 98.93987274169922,
                "Value": true
            },
            "BoundingBox": {
                "Width": 0.12368916720151901,
                "Top": 0.16007372736930847,
                "Left": 0.5901257991790771,
                "Height": 0.25140416622161865
            },
            "Smile": {
                "Confidence": 93.4493179321289,
                "Value": false
            },
            "MouthOpen": {
                "Confidence": 90.53053283691406,
                "Value": false
```

```
            },
            "Quality": {
                "Sharpness": 95.51618957519531,
                "Brightness": 65.29893493652344
            },
            "Mustache": {
                "Confidence": 89.85221099853516,
                "Value": false
            },
            "Beard": {
                "Confidence": 86.1991195678711,
                "Value": true
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Détection de visages dans une image dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectFacesà la section Référence des AWS CLI commandes.

`detect-labels`

L'exemple de code suivant montre comment utiliser`detect-labels`.

Pour plus d'informations, veuillez consulter Détection des étiquettes dans une image.

AWS CLI

Pour détecter une étiquette dans une image

L'`detect-labels`exemple suivant détecte des scènes et des objets dans une image stockée dans un compartiment Amazon S3.

```
aws rekognition detect-labels \
    --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

Sortie :

```
{
    "Labels": [
```

```
        {
            "Instances": [],
            "Confidence": 99.15271759033203,
            "Parents": [
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Automobile"
        },
        {
            "Instances": [],
            "Confidence": 99.15271759033203,
            "Parents": [
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Vehicle"
        },
        {
            "Instances": [],
            "Confidence": 99.15271759033203,
            "Parents": [],
            "Name": "Transportation"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.10616336017847061,
                        "Top": 0.5039216876029968,
                        "Left": 0.0037978808395564556,
                        "Height": 0.18528179824352264
                    },
                    "Confidence": 99.15271759033203
                },
                {
                    "BoundingBox": {
                        "Width": 0.2429988533258438,
                        "Top": 0.5251884460449219,
```

```
                "Left": 0.7309805154800415,
                "Height": 0.21577216684818268
            },
            "Confidence": 99.1286392211914
        },
        {
            "BoundingBox": {
                "Width": 0.14233611524105072,
                "Top": 0.5333095788955688,
                "Left": 0.6494812965393066,
                "Height": 0.15528248250484467
            },
            "Confidence": 98.48368072509766
        },
        {
            "BoundingBox": {
                "Width": 0.11086395382881165,
                "Top": 0.5354844927787781,
                "Left": 0.10355594009160995,
                "Height": 0.10271988064050674
            },
            "Confidence": 96.45606231689453
        },
        {
            "BoundingBox": {
                "Width": 0.06254628300666809,
                "Top": 0.5573825240135193,
                "Left": 0.46083059906959534,
                "Height": 0.053911514580249786
            },
            "Confidence": 93.65448760986328
        },
        {
            "BoundingBox": {
                "Width": 0.10105438530445099,
                "Top": 0.534368634223938,
                "Left": 0.5743985772132874,
                "Height": 0.12226245552301407
            },
            "Confidence": 93.06217193603516
        },
        {
            "BoundingBox": {
                "Width": 0.056389667093753815,
```

```
                "Top": 0.5235804319381714,
                "Left": 0.9427769780158997,
                "Height": 0.17163699865341187
            },
            "Confidence": 92.6864013671875
        },
        {
            "BoundingBox": {
                "Width": 0.06003860384225845,
                "Top": 0.5441341400146484,
                "Left": 0.22409997880458832,
                "Height": 0.06737709045410156
            },
            "Confidence": 90.4227066040039
        },
        {
            "BoundingBox": {
                "Width": 0.02848697081208229,
                "Top": 0.5107086896896362,
                "Left": 0,
                "Height": 0.19150497019290924
            },
            "Confidence": 86.65286254882812
        },
        {
            "BoundingBox": {
                "Width": 0.04067881405353546,
                "Top": 0.5566273927688599,
                "Left": 0.316415935754776,
                "Height": 0.03428703173995018
            },
            "Confidence": 85.36471557617188
        },
        {
            "BoundingBox": {
                "Width": 0.043411049991846085,
                "Top": 0.5394920110702515,
                "Left": 0.18293385207653046,
                "Height": 0.0893595889210701
            },
            "Confidence": 82.21705627441406
        },
        {
            "BoundingBox": {
```

```
                        "Width": 0.031183116137981415,
                        "Top": 0.5579366683959961,
                        "Left": 0.2853088080883026,
                        "Height": 0.03989990055561066
                    },
                    "Confidence": 81.0157470703125
                },
                {
                    "BoundingBox": {
                        "Width": 0.031113790348172188,
                        "Top": 0.5504819750785828,
                        "Left": 0.2580395042896271,
                        "Height": 0.056484755128622055
                    },
                    "Confidence": 56.13441467285156
                },
                {
                    "BoundingBox": {
                        "Width": 0.08586374670267105,
                        "Top": 0.5438792705535889,
                        "Left": 0.5128012895584106,
                        "Height": 0.08550430089235306
                    },
                    "Confidence": 52.37760925292969
                }
            ],
            "Confidence": 99.15271759033203,
            "Parents": [
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Car"
        },
        {
            "Instances": [],
            "Confidence": 98.9914321899414,
            "Parents": [],
            "Name": "Human"
        },
        {
```

```
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.19360728561878204,
                        "Top": 0.35072067379951477,
                        "Left": 0.43734854459762573,
                        "Height": 0.2742200493812561
                    },
                    "Confidence": 98.9914321899414
                },
                {
                    "BoundingBox": {
                        "Width": 0.03801717236638069,
                        "Top": 0.5010883808135986,
                        "Left": 0.9155802130699158,
                        "Height": 0.06597328186035156
                    },
                    "Confidence": 85.02790832519531
                }
            ],
            "Confidence": 98.9914321899414,
            "Parents": [],
            "Name": "Person"
        },
        {
            "Instances": [],
            "Confidence": 93.24951934814453,
            "Parents": [],
            "Name": "Machine"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.03561960905790329,
                        "Top": 0.6468243598937988,
                        "Left": 0.7850857377052307,
                        "Height": 0.08878646790981293
                    },
                    "Confidence": 93.24951934814453
                },
                {
                    "BoundingBox": {
                        "Width": 0.02217046171426773,
```

```
                    "Top": 0.6149078607559204,
                    "Left": 0.04757237061858177,
                    "Height": 0.07136218994855881
                },
                "Confidence": 91.5025863647461
            },
            {
                "BoundingBox": {
                    "Width": 0.016197510063648224,
                    "Top": 0.6274210214614868,
                    "Left": 0.6472989320755005,
                    "Height": 0.04955997318029404
                },
                "Confidence": 85.14686584472656
            },
            {
                "BoundingBox": {
                    "Width": 0.020207518711686134,
                    "Top": 0.6348286867141724,
                    "Left": 0.7295016646385193,
                    "Height": 0.07059963047504425
                },
                "Confidence": 83.34547424316406
            },
            {
                "BoundingBox": {
                    "Width": 0.020280985161662102,
                    "Top": 0.6171894669532776,
                    "Left": 0.08744934946298599,
                    "Height": 0.05297485366463661
                },
                "Confidence": 79.9981460571289
            },
            {
                "BoundingBox": {
                    "Width": 0.018318990245461464,
                    "Top": 0.623889148235321,
                    "Left": 0.6836880445480347,
                    "Height": 0.06730121374130249
                },
                "Confidence": 78.87144470214844
            },
            {
                "BoundingBox": {
```

```
                        "Width": 0.021310249343514442,
                        "Top": 0.6167286038398743,
                        "Left": 0.004064912907779217,
                        "Height": 0.08317798376083374
                    },
                    "Confidence": 75.89361572265625
                },
                {
                    "BoundingBox": {
                        "Width": 0.03604431077837944,
                        "Top": 0.7030032277107239,
                        "Left": 0.9254803657531738,
                        "Height": 0.04569442570209503
                    },
                    "Confidence": 64.402587890625
                },
                {
                    "BoundingBox": {
                        "Width": 0.009834849275648594,
                        "Top": 0.5821820497512817,
                        "Left": 0.28094568848609924,
                        "Height": 0.01964157074689865
                    },
                    "Confidence": 62.79907989501953
                },
                {
                    "BoundingBox": {
                        "Width": 0.01475677452981472,
                        "Top": 0.6137543320655823,
                        "Left": 0.5950819253921509,
                        "Height": 0.039063986390829086
                    },
                    "Confidence": 59.40483474731445
                }
            ],
            "Confidence": 93.24951934814453,
            "Parents": [
                {
                    "Name": "Machine"
                }
            ],
            "Name": "Wheel"
        },
        {
```

```
            "Instances": [],
            "Confidence": 92.61514282226562,
            "Parents": [],
            "Name": "Road"
        },
        {
            "Instances": [],
            "Confidence": 92.37877655029297,
            "Parents": [
                {
                    "Name": "Person"
                }
            ],
            "Name": "Sport"
        },
        {
            "Instances": [],
            "Confidence": 92.37877655029297,
            "Parents": [
                {
                    "Name": "Person"
                }
            ],
            "Name": "Sports"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.12326609343290329,
                        "Top": 0.6332163214683533,
                        "Left": 0.44815489649772644,
                        "Height": 0.058117982000112534
                    },
                    "Confidence": 92.37877655029297
                }
            ],
            "Confidence": 92.37877655029297,
            "Parents": [
                {
                    "Name": "Person"
                },
                {
                    "Name": "Sport"
```

```
                }
            ],
            "Name": "Skateboard"
        },
        {
            "Instances": [],
            "Confidence": 90.62931060791016,
            "Parents": [
                {
                    "Name": "Person"
                }
            ],
            "Name": "Pedestrian"
        },
        {
            "Instances": [],
            "Confidence": 88.81334686279297,
            "Parents": [],
            "Name": "Asphalt"
        },
        {
            "Instances": [],
            "Confidence": 88.81334686279297,
            "Parents": [],
            "Name": "Tarmac"
        },
        {
            "Instances": [],
            "Confidence": 88.23201751708984,
            "Parents": [],
            "Name": "Path"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [],
            "Name": "Urban"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [
                {
                    "Name": "Building"
```

```
                },
                {
                    "Name": "Urban"
                }
            ],
            "Name": "Town"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [],
            "Name": "Building"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                }
            ],
            "Name": "City"
        },
        {
            "Instances": [],
            "Confidence": 78.37934875488281,
            "Parents": [
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Parking Lot"
        },
        {
            "Instances": [],
```

```
            "Confidence": 78.37934875488281,
            "Parents": [
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Parking"
        },
        {
            "Instances": [],
            "Confidence": 74.37590026855469,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                },
                {
                    "Name": "City"
                }
            ],
            "Name": "Downtown"
        },
        {
            "Instances": [],
            "Confidence": 69.84622955322266,
            "Parents": [
                {
                    "Name": "Road"
                }
            ],
            "Name": "Intersection"
        },
        {
            "Instances": [],
            "Confidence": 57.68518829345703,
            "Parents": [
```

```
                {
                    "Name": "Sports Car"
                },
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Coupe"
        },
        {
            "Instances": [],
            "Confidence": 57.68518829345703,
            "Parents": [
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Sports Car"
        },
        {
            "Instances": [],
            "Confidence": 56.59492111206055,
            "Parents": [
                {
                    "Name": "Path"
                }
            ],
            "Name": "Sidewalk"
        },
        {
            "Instances": [],
            "Confidence": 56.59492111206055,
```

```
            "Parents": [
                {
                    "Name": "Path"
                }
            ],
            "Name": "Pavement"
        },
        {
            "Instances": [],
            "Confidence": 55.58770751953125,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                }
            ],
            "Name": "Neighborhood"
        }
    ],
    "LabelModelVersion": "2.0"
}
```

Pour plus d'informations, consultez la section Détection d'étiquettes dans une image dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectLabelsà la section Référence des AWS CLI commandes.

`detect-moderation-labels`

L'exemple de code suivant montre comment utiliser`detect-moderation-labels`.

Pour plus d'informations, veuillez consulter Détecter des images inappropriées.

AWS CLI

Pour détecter le contenu dangereux d'une image

La `detect-moderation-labels` commande suivante détecte le contenu non sécurisé dans l'image spécifiée stockée dans un compartiment Amazon S3.

```
aws rekognition detect-moderation-labels \
    --image "S3Object={Bucket=MyImageS3Bucket,Name=gun.jpg}"
```

Sortie :

```
{
    "ModerationModelVersion": "3.0",
    "ModerationLabels": [
        {
            "Confidence": 97.29618072509766,
            "ParentName": "Violence",
            "Name": "Weapon Violence"
        },
        {
            "Confidence": 97.29618072509766,
            "ParentName": "",
            "Name": "Violence"
        }
    ]
}
```

Pour plus d'informations, consultez la section Détection des images non sécurisées dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DetectModerationLabelsà la section Référence des AWS CLI commandes.

`detect-text`

L'exemple de code suivant montre comment utiliser`detect-text`.

Pour plus d'informations, consultez Détection de texte dans une image.

AWS CLI

Pour détecter du texte dans une image

La `detect-text` commande suivante détecte le texte dans l'image spécifiée.

```
aws rekognition detect-text \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'
```

Sortie :


```
{
    "TextDetections": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.24624845385551453,
                    "Top": 0.28288066387176514,
                    "Left": 0.391388863325119,
                    "Height": 0.022687450051307678
                },
                "Polygon": [
                    {
                        "Y": 0.28288066387176514,
                        "X": 0.391388863325119
                    },
                    {
                        "Y": 0.2826388478279114,
                        "X": 0.6376373171806335
                    },
                    {
                        "Y": 0.30532628297805786,
                        "X": 0.637677013874054
                    },
                    {
                        "Y": 0.305568128824234,
                        "X": 0.39142853021621704
                    }
                ]
            },
            "Confidence": 94.35709381103516,
            "DetectedText": "ESTD 1882",
            "Type": "LINE",
            "Id": 0
        },
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33933889865875244,
                    "Top": 0.32603850960731506,
                    "Left": 0.34534579515457153,
                    "Height": 0.07126858830451965
                },
```

```
                "Polygon": [
                    {
                        "Y": 0.32603850960731506,
                        "X": 0.34534579515457153
                    },
                    {
                        "Y": 0.32633158564567566,
                        "X": 0.684684693813324
                    },
                    {
                        "Y": 0.3976001739501953,
                        "X": 0.684575080871582
                    },
                    {
                        "Y": 0.3973070979118347,
                        "X": 0.345236212015152
                    }
                ]
            },
            "Confidence": 99.95779418945312,
            "DetectedText": "BRAINS",
            "Type": "LINE",
            "Id": 1
        },
        {
            "Confidence": 97.22098541259766,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.061079490929841995,
                    "Top": 0.2843210697174072,
                    "Left": 0.391391396522522,
                    "Height": 0.021029088646173477
                },
                "Polygon": [
                    {
                        "Y": 0.2843210697174072,
                        "X": 0.391391396522522
                    },
                    {
                        "Y": 0.2828207015991211,
                        "X": 0.4524524509906769
                    },
                    {
                        "Y": 0.3038259446620941,
```

```
                        "X": 0.4534534513950348
                    },
                    {
                        "Y": 0.30532634258270264,
                        "X": 0.3923923969268799
                    }
                ]
            },
            "DetectedText": "ESTD",
            "ParentId": 0,
            "Type": "WORD",
            "Id": 2
        },
        {
            "Confidence": 91.49320983886719,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.07007007300853729,
                    "Top": 0.2828207015991211,
                    "Left": 0.5675675868988037,
                    "Height": 0.02250562608242035
                },
                "Polygon": [
                    {
                        "Y": 0.2828207015991211,
                        "X": 0.5675675868988037
                    },
                    {
                        "Y": 0.2828207015991211,
                        "X": 0.6376376152038574
                    },
                    {
                        "Y": 0.30532634258270264,
                        "X": 0.6376376152038574
                    },
                    {
                        "Y": 0.30532634258270264,
                        "X": 0.5675675868988037
                    }
                ]
            },
            "DetectedText": "1882",
            "ParentId": 0,
            "Type": "WORD",
```

```
            "Id": 3
        },
        {
            "Confidence": 99.95779418945312,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33933934569358826,
                    "Top": 0.32633158564567566,
                    "Left": 0.3453453481197357,
                    "Height": 0.07127484679222107
                },
                "Polygon": [
                    {
                        "Y": 0.32633158564567566,
                        "X": 0.3453453481197357
                    },
                    {
                        "Y": 0.32633158564567566,
                        "X": 0.684684693813324
                    },
                    {
                        "Y": 0.39759939908981323,
                        "X": 0.6836836934089661
                    },
                    {
                        "Y": 0.39684921503067017,
                        "X": 0.3453453481197357
                    }
                ]
            },
            "DetectedText": "BRAINS",
            "ParentId": 1,
            "Type": "WORD",
            "Id": 4
        }
    ]
}
```


- Pour plus de détails sur l'API, reportez-vous DetectTextà la section Référence des AWS CLI commandes.

`disassociate-faces`

L'exemple de code suivant montre comment utiliser`disassociate-faces`.

AWS CLI

```
aws rekognition disassociate-faces --face-ids list-of-face-ids
  --user-id user-id --collection-id collection-name --region region-name
```

- Pour plus de détails sur l'API, reportez-vous DisassociateFacesà la section Référence des AWS CLI commandes.

`get-celebrity-info`

L'exemple de code suivant montre comment utiliser`get-celebrity-info`.

AWS CLI

Pour obtenir des informations sur une célébrité

La `get-celebrity-info` commande suivante affiche des informations sur la célébrité spécifiée. Le `id` paramètre provient d'un appel précédent à`recognize-celebrities`.

```
aws rekognition get-celebrity-info --id nnnnnnn
```

Sortie :

```
{
    "Name": "Celeb A",
    "Urls": [
        "www.imdb.com/name/aaaaaaaaa"
    ]
}
```

Pour plus d'informations, consultez la section Obtenir des informations sur une célébrité dans le guide du développeur Amazon Rekognition.

- Pour plus de détails sur l'API, reportez-vous GetCelebrityInfoà la section Référence des AWS CLI commandes.

`get-celebrity-recognition`

L'exemple de code suivant montre comment utiliser`get-celebrity-recognition`.

AWS CLI

Pour obtenir les résultats d'une opération de reconnaissance de célébrités

La `get-celebrity-recognition` commande suivante affiche les résultats d'une opération de reconnaissance de célébrités que vous avez lancée précédemment en appelant`start-celebrity-recognition`.

```
aws rekognition get-celebrity-recognition  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Sortie :

```
{
    "NextToken": "3D01ClxlCiT31VsRDkAO3IybLb/h5AtDWSGuhYi
+N1FIJwwPtAkuKzDhL2rV3GcwmNt77+12",
    "Celebrities": [
        {
            "Timestamp": 0,
            "Celebrity": {
                "Confidence": 96.0,
                "Face": {
                    "BoundingBox": {
                        "Width": 0.70333331823349,
                        "Top": 0.16750000417232513,
                        "Left": 0.19555555284023285,
                        "Height": 0.3956249952316284
                    },
                    "Landmarks": [
                        {
                            "Y": 0.31031012535095215,
                            "X": 0.441436767578125,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.3081788718700409,
                            "X": 0.6437258720397949,
                            "Type": "eyeRight"
                        },
```

```
                    {
                        "Y": 0.39542075991630554,
                        "X": 0.5572493076324463,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.4597957134246826,
                        "X": 0.4579732120037079,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.45688048005104065,
                        "X": 0.6349081993103027,
                        "Type": "mouthRight"
                    }
                ],
                "Pose": {
                    "Yaw": 8.943398475646973,
                    "Roll": -2.0309247970581055,
                    "Pitch": -0.5674862861633301
                },
                "Quality": {
                    "Sharpness": 99.40211486816406,
                    "Brightness": 89.47132110595703
                },
                "Confidence": 99.99861145019531
            },
            "Name": "CelebrityA",
            "Urls": [
                "www.imdb.com/name/111111111"
            ],
            "Id": "nnnnnn"
        }
    },
    {
        "Timestamp": 467,
        "Celebrity": {
            "Confidence": 99.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.6877777576446533,
                    "Top": 0.18437500298023224,
                    "Left": 0.20555555820465088,
                    "Height": 0.3868750035762787
```

```
                    },
                    "Landmarks": [
                        {
                            "Y": 0.31895750761032104,
                            "X": 0.4411413371562958,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.3140959143638611,
                            "X": 0.6523157954216003,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.4016456604003906,
                            "X": 0.5682755708694458,
                            "Type": "nose"
                        },
                        {
                            "Y": 0.46894142031669617,
                            "X": 0.4597797095775604,
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.46971091628074646,
                            "X": 0.6286435127258301,
                            "Type": "mouthRight"
                        }
                    ],
                    "Pose": {
                        "Yaw": 10.433465957641602,
                        "Roll": -3.347442388534546,
                        "Pitch": 1.3709543943405151
                    },
                    "Quality": {
                        "Sharpness": 99.5531005859375,
                        "Brightness": 88.5764389038086
                    },
                    "Confidence": 99.99148559570312
                },
                "Name": "Jane Celebrity",
                "Urls": [
                    "www.imdb.com/name/111111111"
                ],
                "Id": "nnnnnn"
```

```
            }
        }
    ],
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.978118896484375,
        "Codec": "h264",
        "DurationMillis": 4570,
        "FrameHeight": 1920,
        "FrameWidth": 1080
    }
}
```

Pour plus d'informations, consultez la section Reconnaître les célébrités dans une vidéo stockée dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetCelebrityRecognitionà la section Référence des AWS CLI commandes.

`get-content-moderation`

L'exemple de code suivant montre comment utiliser`get-content-moderation`.

AWS CLI

Pour obtenir les résultats d'une opération de contenu non sécurisé

La `get-content-moderation` commande suivante affiche les résultats d'une opération de contenu non sécurisé que vous avez lancée précédemment en appelant`start-content-moderation`.

```
aws rekognition get-content-moderation \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Sortie :

```
{
    "NextToken": "dlhcKMHMzpCBGFukz6IO3JMcWiJAamCVhXHt3r6b4b5Tfbyw3q7o+Jeezt
+ZpgfOnW9FCCgQ",
    "ModerationLabels": [
        {
```

```
            "Timestamp": 0,
            "ModerationLabel": {
                "Confidence": 97.39583587646484,
                "ParentName": "",
                "Name": "Violence"
            }
        },
        {
            "Timestamp": 0,
            "ModerationLabel": {
                "Confidence": 97.39583587646484,
                "ParentName": "Violence",
                "Name": "Weapon Violence"
            }
        }
    ],
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.97515869140625,
        "Codec": "h264",
        "DurationMillis": 6039,
        "FrameHeight": 1920,
        "FrameWidth": 1080
    }
}
```


Pour plus d'informations, consultez la section Détection des vidéos stockées non sécurisées dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetContentModerationà la section Référence des AWS CLI commandes.

`get-face-detection`

L'exemple de code suivant montre comment utiliser`get-face-detection`.

AWS CLI

Pour obtenir les résultats d'une opération de détection de visages

La `get-face-detection` commande suivante affiche les résultats d'une opération de détection de visages que vous avez lancée précédemment en appelant`start-face-detection`.

```
aws rekognition get-face-detection \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Sortie :


```
{
    "Faces": [
        {
            "Timestamp": 467,
            "Face": {
                "BoundingBox": {
                    "Width": 0.1560753583908081,
                    "Top": 0.13555361330509186,
                    "Left": -0.0952017530798912,
                    "Height": 0.6934483051300049
                },
                "Landmarks": [
                    {
                        "Y": 0.4013825058937073,
                        "X": -0.041750285774469376,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.41695496439933777,
                        "X": 0.027979329228401184,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.6375303268432617,
                        "X": -0.04034662991762161,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.6497718691825867,
                        "X": 0.013960429467260838,
                        "Type": "mouthRight"
                    },
                    {
                        "Y": 0.5238034129142761,
                        "X": 0.008022055961191654,
                        "Type": "nose"
                    }
                ],
```

```
                "Pose": {
                    "Yaw": -58.07863998413086,
                    "Roll": 1.9384294748306274,
                    "Pitch": -24.66305160522461
                },
                "Quality": {
                    "Sharpness": 83.14741516113281,
                    "Brightness": 25.75942611694336
                },
                "Confidence": 87.7622299194336
            }
        },
        {
            "Timestamp": 967,
            "Face": {
                "BoundingBox": {
                    "Width": 0.28559377789497375,
                    "Top": 0.19436298310756683,
                    "Left": 0.024553587660193443,
                    "Height": 0.7216082215309143
                },
                "Landmarks": [
                    {
                        "Y": 0.4650231599807739,
                        "X": 0.16269078850746155,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.4843238294124603,
                        "X": 0.2782580852508545,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.71530681848526,
                        "X": 0.1741468608379364,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.7310671210289001,
                        "X": 0.26857468485832214,
                        "Type": "mouthRight"
                    },
                    {
                        "Y": 0.582602322101593,
```

```
                        "X": 0.2566150426864624,
                        "Type": "nose"
                    }
                ],
                "Pose": {
                    "Yaw": 11.487052917480469,
                    "Roll": 5.074230670928955,
                    "Pitch": 15.396159172058105
                },
                "Quality": {
                    "Sharpness": 73.32209777832031,
                    "Brightness": 54.96497344970703
                },
                "Confidence": 99.99998474121094
            }
        }
    ],
    "NextToken":
 "OzL223pDKy9116O/02KXRqFIEAwxjy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```


Pour plus d'informations, consultez la section Détection de visages dans une vidéo stockée dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetFaceDetectionà la section Référence des AWS CLI commandes.

`get-face-search`

L'exemple de code suivant montre comment utiliser`get-face-search`.

AWS CLI

Pour obtenir les résultats d'une opération de recherche faciale

La `get-face-search` commande suivante affiche les résultats d'une opération de recherche faciale que vous avez lancée précédemment en appelant`start-face-search`.

```
aws rekognition get-face-search  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Sortie :

```
{
    "Persons": [
        {
            "Timestamp": 467,
            "FaceMatches": [],
            "Person": {
                "Index": 0,
                "Face": {
                    "BoundingBox": {
                        "Width": 0.1560753583908081,
                        "Top": 0.13555361330509186,
                        "Left": -0.0952017530798912,
                        "Height": 0.6934483051300049
                    },
                    "Landmarks": [
                        {
                            "Y": 0.4013825058937073,
                            "X": -0.041750285774469376,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.41695496439933777,
                            "X": 0.027979329228401184,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.6375303268432617,
                            "X": -0.04034662991762161,
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.6497718691825867,
                            "X": 0.013960429467260838,
                            "Type": "mouthRight"
                        },
```

```
                    {
                        "Y": 0.5238034129142761,
                        "X": 0.008022055961191654,
                        "Type": "nose"
                    }
                ],
                "Pose": {
                    "Yaw": -58.07863998413086,
                    "Roll": 1.9384294748306274,
                    "Pitch": -24.66305160522461
                },
                "Quality": {
                    "Sharpness": 83.14741516113281,
                    "Brightness": 25.75942611694336
                },
                "Confidence": 87.7622299194336
            }
        }
    },
    {
        "Timestamp": 967,
        "FaceMatches": [
            {
                "Face": {
                    "BoundingBox": {
                        "Width": 0.12368900328874588,
                        "Top": 0.16007399559020996,
                        "Left": 0.5901259779930115,
                        "Height": 0.2514039874076843
                    },
                    "FaceId": "056a95fa-2060-4159-9cab-7ed4daa030fa",
                    "ExternalImageId": "image3.jpg",
                    "Confidence": 100.0,
                    "ImageId": "08f8a078-8929-37fd-8e8f-aadf690e8232"
                },
                "Similarity": 98.44476318359375
            }
        ],
        "Person": {
            "Index": 1,
            "Face": {
                "BoundingBox": {
                    "Width": 0.28559377789497375,
                    "Top": 0.19436298310756683,
```

```
                        "Left": 0.024553587660193443,
                        "Height": 0.7216082215309143
                    },
                    "Landmarks": [
                        {
                            "Y": 0.4650231599807739,
                            "X": 0.16269078850746155,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.4843238294124603,
                            "X": 0.2782580852508545,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.71530681848526,
                            "X": 0.1741468608379364,
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.7310671210289001,
                            "X": 0.26857468485832214,
                            "Type": "mouthRight"
                        },
                        {
                            "Y": 0.582602322101593,
                            "X": 0.2566150426864624,
                            "Type": "nose"
                        }
                    ],
                    "Pose": {
                        "Yaw": 11.487052917480469,
                        "Roll": 5.074230670928955,
                        "Pitch": 15.396159172058105
                    },
                    "Quality": {
                        "Sharpness": 73.32209777832031,
                        "Brightness": 54.96497344970703
                    },
                    "Confidence": 99.99998474121094
                }
            }
        }
    ],
```

```
    "NextToken": "5bkgcezyuaqhtWk3C8OTW6cjRghrwV9XDMivm5B3MXm+Lv6G+L+GejyFHPhoNa/
ldXIC4c/d",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```

Pour plus d'informations, consultez la section Rechercher des visages dans les vidéos stockées dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetFaceSearchà la section Référence des AWS CLI commandes.

`get-label-detection`

L'exemple de code suivant montre comment utiliser`get-label-detection`.

AWS CLI

Pour obtenir les résultats d'une opération de détection d'objets et de scènes

La `get-label-detection` commande suivante affiche les résultats d'une opération de détection d'objets et de scènes que vous avez lancée précédemment en appelant`start-label-detection`.

```
aws rekognition get-label-detection  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Sortie :

```
{
    "Labels": [
        {
            "Timestamp": 0,
            "Label": {
                "Instances": [],
```

```
                "Confidence": 50.19071578979492,
                "Parents": [
                    {
                        "Name": "Person"
                    },
                    {
                        "Name": "Crowd"
                    }
                ],
                "Name": "Audience"
            }
        },
        {
            "Timestamp": 0,
            "Label": {
                "Instances": [],
                "Confidence": 55.74115753173828,
                "Parents": [
                    {
                        "Name": "Room"
                    },
                    {
                        "Name": "Indoors"
                    },
                    {
                        "Name": "School"
                    }
                ],
                "Name": "Classroom"
            }
        }
    ],
    "JobStatus": "SUCCEEDED",
    "LabelModelVersion": "2.0",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    },
    "NextToken": "BMugzAi4L72IERzQdbpyMQuEFBsjlo5W0Yx3mfG+sR9mm98E1/
CpObenspRfs/5FBQFs4X7G"
```

```
}
```

Pour plus d'informations, consultez la section Détection d'étiquettes dans une vidéo du manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetLabelDetectionà la section Référence des AWS CLI commandes.

`get-person-tracking`

L'exemple de code suivant montre comment utiliser`get-person-tracking`.

AWS CLI

Pour obtenir les résultats d'une opération de repérage des personnes

La `get-person-tracking` commande suivante affiche les résultats d'une opération de repérage de personnes que vous avez lancée précédemment en appelant`start-person-tracking`.

```
aws rekognition get-person-tracking  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Sortie :

```
{
    "Persons": [
        {
            "Timestamp": 500,
            "Person": {
                "BoundingBox": {
                    "Width": 0.4151041805744171,
                    "Top": 0.07870370149612427,
                    "Left": 0.0,
                    "Height": 0.9212962985038757
                },
                "Index": 0
            }
        },
        {
            "Timestamp": 567,
```

```
            "Person": {
                "BoundingBox": {
                    "Width": 0.4755208194255829,
                    "Top": 0.07777778059244156,
                    "Left": 0.0,
                    "Height": 0.9194444417953491
                },
                "Index": 0
            }
        }
    ],
    "NextToken": "D/vRIYNyhG79ugdta3f+8cRg9oSRo
+HigGOuxRiYpTn0ExnqTi1CJektVAc4HrAXDv25eHYk",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```

Pour plus d'informations, consultez People Pathing dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetPersonTrackingà la section Référence des AWS CLI commandes.

`index-faces`

L'exemple de code suivant montre comment utiliser`index-faces`.

Pour plus d'informations, veuillez consulter Ajouter des visages à une collection.

AWS CLI

Pour ajouter des visages à une collection

La `index-faces` commande suivante ajoute les visages trouvés dans une image à la collection spécifiée.

```
aws rekognition index-faces \
    --image '{"S3Object":{"Bucket":"MyVideoS3Bucket","Name":"MyPicture.jpg"}}' \
    --collection-id MyCollection \
    --max-faces 1 \
    --quality-filter "AUTO" \
    --detection-attributes "ALL" \
    --external-image-id "MyPicture.jpg"
```

Sortie :


```
{
    "FaceRecords": [
        {
            "FaceDetail": {
                "Confidence": 99.993408203125,
                "Eyeglasses": {
                    "Confidence": 99.11750030517578,
                    "Value": false
                },
                "Sunglasses": {
                    "Confidence": 99.98249053955078,
                    "Value": false
                },
                "Gender": {
                    "Confidence": 99.92769622802734,
                    "Value": "Male"
                },
                "Landmarks": [
                    {
                        "Y": 0.26750367879867554,
                        "X": 0.6202793717384338,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.26642778515815735,
                        "X": 0.6787431836128235,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.31361380219459534,
                        "X": 0.6421601176261902,
                        "Type": "nose"
                    },
```

```
                    {
                        "Y": 0.3495299220085144,
                        "X": 0.6216195225715637,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.35194727778434753,
                        "X": 0.669899046421051,
                        "Type": "mouthRight"
                    },
                    {
                        "Y": 0.26844894886016846,
                        "X": 0.6210268139839172,
                        "Type": "leftPupil"
                    },
                    {
                        "Y": 0.26707562804222107,
                        "X": 0.6817160844802856,
                        "Type": "rightPupil"
                    },
                    {
                        "Y": 0.24834522604942322,
                        "X": 0.6018546223640442,
                        "Type": "leftEyeBrowLeft"
                    },
                    {
                        "Y": 0.24397172033786774,
                        "X": 0.6172008514404297,
                        "Type": "leftEyeBrowUp"
                    },
                    {
                        "Y": 0.24677404761314392,
                        "X": 0.6339119076728821,
                        "Type": "leftEyeBrowRight"
                    },
                    {
                        "Y": 0.24582654237747192,
                        "X": 0.6619398593902588,
                        "Type": "rightEyeBrowLeft"
                    },
                    {
                        "Y": 0.23973053693771362,
                        "X": 0.6804757118225098,
                        "Type": "rightEyeBrowUp"
```

```
                    },
                    {
                        "Y": 0.24441994726657867,
                        "X": 0.6978968977928162,
                        "Type": "rightEyeBrowRight"
                    },
                    {
                        "Y": 0.2695908546447754,
                        "X": 0.6085202693939209,
                        "Type": "leftEyeLeft"
                    },
                    {
                        "Y": 0.26716896891593933,
                        "X": 0.6315826177597046,
                        "Type": "leftEyeRight"
                    },
                    {
                        "Y": 0.26289820671081543,
                        "X": 0.6202316880226135,
                        "Type": "leftEyeUp"
                    },
                    {
                        "Y": 0.27123287320137024,
                        "X": 0.6205548048019409,
                        "Type": "leftEyeDown"
                    },
                    {
                        "Y": 0.2668408751487732,
                        "X": 0.6663622260093689,
                        "Type": "rightEyeLeft"
                    },
                    {
                        "Y": 0.26741549372673035,
                        "X": 0.6910083889961243,
                        "Type": "rightEyeRight"
                    },
                    {
                        "Y": 0.2614026665687561,
                        "X": 0.6785826086997986,
                        "Type": "rightEyeUp"
                    },
                    {
                        "Y": 0.27075251936912537,
                        "X": 0.6789616942405701,
```

```
                        "Type": "rightEyeDown"
                    },
                    {
                        "Y": 0.3211299479007721,
                        "X": 0.6324167847633362,
                        "Type": "noseLeft"
                    },
                    {
                        "Y": 0.32276326417922974,
                        "X": 0.6558475494384766,
                        "Type": "noseRight"
                    },
                    {
                        "Y": 0.34385165572166443,
                        "X": 0.6444970965385437,
                        "Type": "mouthUp"
                    },
                    {
                        "Y": 0.3671635091304779,
                        "X": 0.6459195017814636,
                        "Type": "mouthDown"
                    }
                ],
                "Pose": {
                    "Yaw": -9.54541015625,
                    "Roll": -0.5709401965141296,
                    "Pitch": 0.6045494675636292
                },
                "Emotions": [
                    {
                        "Confidence": 39.90074157714844,
                        "Type": "HAPPY"
                    },
                    {
                        "Confidence": 23.38753890991211,
                        "Type": "CALM"
                    },
                    {
                        "Confidence": 5.840933322906494,
                        "Type": "CONFUSED"
                    }
                ],
                "AgeRange": {
                    "High": 63,
```

```
                    "Low": 45
                },
                "EyesOpen": {
                    "Confidence": 99.80887603759766,
                    "Value": true
                },
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618015021085739,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770642817020416
                },
                "Smile": {
                    "Confidence": 99.69740295410156,
                    "Value": false
                },
                "MouthOpen": {
                    "Confidence": 99.97393798828125,
                    "Value": false
                },
                "Quality": {
                    "Sharpness": 95.54405975341797,
                    "Brightness": 63.867706298828125
                },
                "Mustache": {
                    "Confidence": 97.05007934570312,
                    "Value": false
                },
                "Beard": {
                    "Confidence": 87.34505462646484,
                    "Value": false
                }
            },
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618015021085739,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770642817020416
                },
                "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
                "ExternalImageId": "example-image.jpg",
                "Confidence": 99.993408203125,
                "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
```

```
            }
        }
    ],
    "UnindexedFaces": [],
    "FaceModelVersion": "3.0",
    "OrientationCorrection": "ROTATE_0"
}
```

Pour plus d'informations, consultez la section Ajouter des visages à une collection dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous IndexFacesà la section Référence des AWS CLI commandes.

`list-collections`

L'exemple de code suivant montre comment utiliser`list-collections`.

Pour en savoir plus, consultez Répertoriage de collections.

AWS CLI

Pour répertorier les collections disponibles

La `list-collections` commande suivante répertorie les collections disponibles dans le AWS compte.

```
aws rekognition list-collections
```

Sortie :

```
{
    "FaceModelVersions": [
        "2.0",
        "3.0",
        "3.0",
        "3.0",
        "4.0",
        "1.0",
        "3.0",
        "4.0",
        "4.0",
        "4.0"
```

```
    ],
    "CollectionIds": [
        "MyCollection1",
        "MyCollection2",
        "MyCollection3",
        "MyCollection4",
        "MyCollection5",
        "MyCollection6",
        "MyCollection7",
        "MyCollection8",
        "MyCollection9",
        "MyCollection10"
    ]
}
```

Pour plus d'informations, consultez Listing Collections dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListCollectionsà la section Référence des AWS CLI commandes.

`list-faces`

L'exemple de code suivant montre comment utiliser`list-faces`.

Pour plus d'informations, consultez Répertoriage de visages d'une collection.

AWS CLI

Pour répertorier les visages d'une collection

La `list-faces` commande suivante répertorie les visages de la collection spécifiée.

```
aws rekognition list-faces \
    --collection-id MyCollection
```

Sortie :

```
{
    "FaceModelVersion": "3.0",
    "Faces": [
        {
            "BoundingBox": {
```

```
                "Width": 0.5216310024261475,
                "Top": 0.3256250023841858,
                "Left": 0.13394300639629364,
                "Height": 0.3918749988079071
            },
            "FaceId": "0040279c-0178-436e-b70a-e61b074e96b0",
            "ExternalImageId": "image1.jpg",
            "Confidence": 100.0,
            "ImageId": "f976e487-3719-5e2d-be8b-ea2724c26991"
        },
        {
            "BoundingBox": {
                "Width": 0.5074880123138428,
                "Top": 0.3774999976158142,
                "Left": 0.18302799761295319,
                "Height": 0.3812499940395355
            },
            "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
            "ExternalImageId": "image2.jpg",
            "Confidence": 99.99930572509766,
            "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
        },
        {
            "BoundingBox": {
                "Width": 0.5574039816856384,
                "Top": 0.37187498807907104,
                "Left": 0.14559100568294525,
                "Height": 0.4181250035762787
            },
            "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
            "ExternalImageId": "image3.jpg",
            "Confidence": 99.99960327148438,
            "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
        },
        {
            "BoundingBox": {
                "Width": 0.18562500178813934,
                "Top": 0.1618019938468933,
                "Left": 0.5575000047683716,
                "Height": 0.24770599603652954
            },
            "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
            "ExternalImageId": "image4.jpg",
            "Confidence": 99.99340057373047,
```

```
            "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
        },
        {
            "BoundingBox": {
                "Width": 0.5307819843292236,
                "Top": 0.2862499952316284,
                "Left": 0.1564060002565384,
                "Height": 0.3987500071525574
            },
            "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
            "ExternalImageId": "image5.jpg",
            "Confidence": 99.99970245361328,
            "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
        },
        {
            "BoundingBox": {
                "Width": 0.5773710012435913,
                "Top": 0.34437501430511475,
                "Left": 0.12396000325679779,
                "Height": 0.4337500035762787
            },
            "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
            "ExternalImageId": "image6.jpg",
            "Confidence": 100.0,
            "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
        },
        {
            "BoundingBox": {
                "Width": 0.5349419713020325,
                "Top": 0.29124999046325684,
                "Left": 0.16389399766921997,
                "Height": 0.40187498927116394
            },
            "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
            "ExternalImageId": "image7.jpg",
            "Confidence": 99.99979400634766,
            "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
        },
        {
            "BoundingBox": {
                "Width": 0.41499999165534973,
                "Top": 0.09187500178813934,
                "Left": 0.28083300590515137,
                "Height": 0.3112500011920929
```

```
            },
            "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
            "ExternalImageId": "image8.jpg",
            "Confidence": 99.99769592285156,
            "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
        },
        {
            "BoundingBox": {
                "Width": 0.48166701197624207,
                "Top": 0.20999999344348907,
                "Left": 0.21250000596046448,
                "Height": 0.36125001311302185
            },
            "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
            "ExternalImageId": "image9.jpg",
            "Confidence": 99.99949645996094,
            "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
        },
        {
            "BoundingBox": {
                "Width": 0.18562500178813934,
                "Top": 0.1618019938468933,
                "Left": 0.5575000047683716,
                "Height": 0.24770599603652954
            },
            "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
            "ExternalImageId": "image10.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
        }
    ]
}
```

Pour plus d'informations, consultez la section Listing Faces in a Collection dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListFacesà la section Référence des AWS CLI commandes.

`list-stream-processors`

L'exemple de code suivant montre comment utiliser`list-stream-processors`.

AWS CLI

Pour répertorier les processeurs de streaming de votre compte

La `list-stream-processors` commande suivante répertorie les processeurs de flux de votre compte et leur état.

```
aws rekognition list-stream-processors
```

Sortie :

```
{
    "StreamProcessors": [
        {
            "Status": "STOPPED",
            "Name": "my-stream-processor"
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de vidéos en streaming dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListStreamProcessorsà la section Référence des AWS CLI commandes.

`recognize-celebrities`

L'exemple de code suivant montre comment utiliser`recognize-celebrities`.

Pour plus d'informations, consultez Reconnaissance de célébrités dans une image.

AWS CLI

Reconnaître les célébrités sur une image

La `recognize-celebrities` commande suivante reconnaît les célébrités présentes dans l'image spécifiée stockée dans un compartiment Amazon S3. :

```
aws rekognition recognize-celebrities \
    --image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

Sortie :


```
{
    "UnrecognizedFaces": [
        {
            "BoundingBox": {
                "Width": 0.14416666328907013,
                "Top": 0.07777778059244156,
                "Left": 0.625,
                "Height": 0.2746031880378723
            },
            "Confidence": 99.9990234375,
            "Pose": {
                "Yaw": 10.80408763885498,
                "Roll": -12.761146545410156,
                "Pitch": 10.96889877319336
            },
            "Quality": {
                "Sharpness": 94.1185531616211,
                "Brightness": 79.18367004394531
            },
            "Landmarks": [
                {
                    "Y": 0.18220913410186768,
                    "X": 0.6702951788902283,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.16337193548679352,
                    "X": 0.7188183665275574,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.20739148557186127,
                    "X": 0.7055801749229431,
                    "Type": "nose"
                },
                {
                    "Y": 0.2889308035373688,
                    "X": 0.687512218952179,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.2706988751888275,
```

```
                    "X": 0.7250053286552429,
                    "Type": "mouthRight"
                }
            ]
        }
    ],
    "CelebrityFaces": [
        {
            "MatchConfidence": 100.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.14000000059604645,
                    "Top": 0.1190476194024086,
                    "Left": 0.82833331823349,
                    "Height": 0.2666666805744171
                },
                "Confidence": 99.99359130859375,
                "Pose": {
                    "Yaw": -10.509642601013184,
                    "Roll": -14.51749324798584,
                    "Pitch": 13.799399375915527
                },
                "Quality": {
                    "Sharpness": 78.74752044677734,
                    "Brightness": 42.201324462890625
                },
                "Landmarks": [
                    {
                        "Y": 0.2290833294391632,
                        "X": 0.8709492087364197,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.20639978349208832,
                        "X": 0.9153988361358643,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.25417643785476685,
                        "X": 0.8907724022865295,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.32729196548461914,
```

```
                    "X": 0.8876466155052185,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.3115464746952057,
                    "X": 0.9238573312759399,
                    "Type": "mouthRight"
                }
            ]
        },
        "Name": "Celeb A",
        "Urls": [
            "www.imdb.com/name/aaaaaaaaa"
        ],
        "Id": "1111111"
    },
    {
        "MatchConfidence": 97.0,
        "Face": {
            "BoundingBox": {
                "Width": 0.13333334028720856,
                "Top": 0.24920634925365448,
                "Left": 0.4449999928474426,
                "Height": 0.2539682686328888
            },
            "Confidence": 99.99979400634766,
            "Pose": {
                "Yaw": 6.557040691375732,
                "Roll": -7.316643714904785,
                "Pitch": 9.272967338562012
            },
            "Quality": {
                "Sharpness": 83.23492431640625,
                "Brightness": 78.83267974853516
            },
            "Landmarks": [
                {
                    "Y": 0.3625510632991791,
                    "X": 0.48898839950561523,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.35366007685661316,
                    "X": 0.5313721299171448,
```

```
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.3894785940647125,
                    "X": 0.5173314809799194,
                    "Type": "nose"
                },
                {
                    "Y": 0.44889405369758606,
                    "X": 0.5020005702972412,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.4408611059188843,
                    "X": 0.5351271629333496,
                    "Type": "mouthRight"
                }
            ]
        },
        "Name": "Celeb B",
        "Urls": [
            "www.imdb.com/name/bbbbbbbbb"
        ],
        "Id": "2222222"
    },
    {
        "MatchConfidence": 100.0,
        "Face": {
            "BoundingBox": {
                "Width": 0.12416666746139526,
                "Top": 0.2968254089355469,
                "Left": 0.2150000035762787,
                "Height": 0.23650793731212616
            },
            "Confidence": 99.99958801269531,
            "Pose": {
                "Yaw": 7.801797866821289,
                "Roll": -8.326810836791992,
                "Pitch": 7.844768047332764
            },
            "Quality": {
                "Sharpness": 86.93206024169922,
                "Brightness": 79.81291198730469
            },
```

```
                "Landmarks": [
                    {
                        "Y": 0.4027804136276245,
                        "X": 0.2575301229953766,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.3934555947780609,
                        "X": 0.2956969439983368,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.4309830069541931,
                        "X": 0.2837020754814148,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.48186683654785156,
                        "X": 0.26812544465065,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.47338807582855225,
                        "X": 0.29905644059181213,
                        "Type": "mouthRight"
                    }
                ]
            },
            "Name": "Celeb C",
            "Urls": [
                "www.imdb.com/name/ccccccccc"
            ],
            "Id": "3333333"
        },
        {
            "MatchConfidence": 97.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.11916666477918625,
                    "Top": 0.3698412775993347,
                    "Left": 0.008333333767950535,
                    "Height": 0.22698412835597992
                },
                "Confidence": 99.99999237060547,
```

```
                "Pose": {
                    "Yaw": 16.38478660583496,
                    "Roll": -1.0260354280471802,
                    "Pitch": 5.975185394287109
                },
                "Quality": {
                    "Sharpness": 83.23492431640625,
                    "Brightness": 61.408443450927734
                },
                "Landmarks": [
                    {
                        "Y": 0.4632347822189331,
                        "X": 0.049406956881284714,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.46388113498687744,
                        "X": 0.08722897619009018,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.5020678639411926,
                        "X": 0.0758260041475296,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.544157862663269,
                        "X": 0.054029736667871475,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.5463630557060242,
                        "X": 0.08464983850717545,
                        "Type": "mouthRight"
                    }
                ]
            },
            "Name": "Celeb D",
            "Urls": [
                "www.imdb.com/name/ddddddddd"
            ],
            "Id": "4444444"
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Reconnaître les célébrités sur une image dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RecognizeCelebritiesà la section Référence des AWS CLI commandes.

`search-faces-by-image`

L'exemple de code suivant montre comment utiliser`search-faces-by-image`.

Pour plus d'informations, voir Recherche d'un visage (image).

AWS CLI

Pour rechercher des visages dans une collection qui correspondent au plus grand visage d'une image.

La `search-faces-by-image` commande suivante recherche les visages d'une collection qui correspondent au plus grand visage de l'image spécifiée. :

```
aws rekognition search-faces-by-image \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePerson.jpg"}}' \
    --collection-id MyFaceImageCollection

{
    "SearchedFaceBoundingBox": {
        "Width": 0.18562500178813934,
        "Top": 0.1618015021085739,
        "Left": 0.5575000047683716,
        "Height": 0.24770642817020416
    },
    "SearchedFaceConfidence": 99.993408203125,
    "FaceMatches": [
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618019938468933,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770599603652954
                },
```

```
            "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
            "ExternalImageId": "example-image.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
        },
        "Similarity": 99.97913360595703
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.18562500178813934,
                "Top": 0.1618019938468933,
                "Left": 0.5575000047683716,
                "Height": 0.24770599603652954
            },
            "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
            "ExternalImageId": "image3.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
        },
        "Similarity": 99.97913360595703
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.41499999165534973,
                "Top": 0.09187500178813934,
                "Left": 0.28083300590515137,
                "Height": 0.3112500011920929
            },
            "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
            "ExternalImageId": "image2.jpg",
            "Confidence": 99.99769592285156,
            "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
        },
        "Similarity": 99.18069458007812
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.48166701197624207,
                "Top": 0.20999999344348907,
                "Left": 0.21250000596046448,
                "Height": 0.36125001311302185
```

```
            },
            "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
            "ExternalImageId": "image1.jpg",
            "Confidence": 99.99949645996094,
            "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
        },
        "Similarity": 98.66607666015625
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.5349419713020325,
                "Top": 0.29124999046325684,
                "Left": 0.16389399766921997,
                "Height": 0.40187498927116394
            },
            "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
            "ExternalImageId": "image9.jpg",
            "Confidence": 99.99979400634766,
            "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
        },
        "Similarity": 98.24278259277344
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.5307819843292236,
                "Top": 0.2862499952316284,
                "Left": 0.1564060002565384,
                "Height": 0.3987500071525574
            },
            "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
            "ExternalImageId": "image10.jpg",
            "Confidence": 99.99970245361328,
            "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
        },
        "Similarity": 98.10665893554688
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.5074880123138428,
                "Top": 0.3774999976158142,
                "Left": 0.18302799761295319,
```

```
                    "Height": 0.3812499940395355
                },
                "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
                "ExternalImageId": "image6.jpg",
                "Confidence": 99.99930572509766,
                "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
            },
            "Similarity": 98.10526275634766
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5574039816856384,
                    "Top": 0.37187498807907104,
                    "Left": 0.14559100568294525,
                    "Height": 0.4181250035762787
                },
                "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
                "ExternalImageId": "image5.jpg",
                "Confidence": 99.99960327148438,
                "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
            },
            "Similarity": 97.94659423828125
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5773710012435913,
                    "Top": 0.34437501430511475,
                    "Left": 0.12396000325679779,
                    "Height": 0.4337500035762787
                },
                "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
                "ExternalImageId": "image8.jpg",
                "Confidence": 100.0,
                "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
            },
            "Similarity": 97.93476867675781
        }
    ],

 "FaceModelVersion": "3.0"
}
```

Pour plus d'informations, consultez la section Rechercher un visage à l'aide d'une image dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SearchFacesByImageà la section Référence des AWS CLI commandes.

`search-faces`

L'exemple de code suivant montre comment utiliser`search-faces`.

Pour plus d'informations, veuillez consulter Recherche d'un visage (identification faciale).

AWS CLI

Pour rechercher des visages dans une collection qui correspondent à un identifiant de visage.

La `search-faces` commande suivante recherche les visages d'une collection qui correspondent à l'ID de visage spécifié.

```
aws rekognition search-faces \
    --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \
    --collection-id MyCollection
```

Sortie :

```
{
    "SearchedFaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
    "FaceModelVersion": "3.0",
    "FaceMatches": [
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.48166701197624207,
                    "Top": 0.20999999344348907,
                    "Left": 0.21250000596046448,
                    "Height": 0.36125001311302185
                },
                "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
                "ExternalImageId": "image1.jpg",
                "Confidence": 99.99949645996094,
                "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
            },
            "Similarity": 99.30997467041016
```

```
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618019938468933,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770599603652954
                },
                "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
                "ExternalImageId": "example-image.jpg",
                "Confidence": 99.99340057373047,
                "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
            },
            "Similarity": 99.24862670898438
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618019938468933,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770599603652954
                },
                "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
                "ExternalImageId": "image3.jpg",
                "Confidence": 99.99340057373047,
                "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
            },
            "Similarity": 99.24862670898438
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5349419713020325,
                    "Top": 0.29124999046325684,
                    "Left": 0.16389399766921997,
                    "Height": 0.40187498927116394
                },
                "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
                "ExternalImageId": "image9.jpg",
                "Confidence": 99.99979400634766,
                "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
            },
```

```
            "Similarity": 96.73158264160156
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5307819843292236,
                    "Top": 0.2862499952316284,
                    "Left": 0.1564060002565384,
                    "Height": 0.3987500071525574
                },
                "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
                "ExternalImageId": "image10.jpg",
                "Confidence": 99.99970245361328,
                "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
            },
            "Similarity": 96.48291015625
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5074880123138428,
                    "Top": 0.3774999976158142,
                    "Left": 0.18302799761295319,
                    "Height": 0.3812499940395355
                },
                "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
                "ExternalImageId": "image6.jpg",
                "Confidence": 99.99930572509766,
                "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
            },
            "Similarity": 96.43287658691406
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5574039816856384,
                    "Top": 0.37187498807907104,
                    "Left": 0.14559100568294525,
                    "Height": 0.4181250035762787
                },
                "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
                "ExternalImageId": "image5.jpg",
                "Confidence": 99.99960327148438,
                "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
```

```
            },
            "Similarity": 95.25305938720703
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5773710012435913,
                    "Top": 0.34437501430511475,
                    "Left": 0.12396000325679779,
                    "Height": 0.4337500035762787
                },
                "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
                "ExternalImageId": "image8.jpg",
                "Confidence": 100.0,
                "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
            },
            "Similarity": 95.22837829589844
        }
    ]
}
```

Pour plus d'informations, consultez la section Rechercher un visage à l'aide de son identifiant facial dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SearchFacesà la section Référence des AWS CLI commandes.

`start-celebrity-recognition`

L'exemple de code suivant montre comment utiliser`start-celebrity-recognition`.

AWS CLI

Pour démarrer la reconnaissance de célébrités dans une vidéo enregistrée

La `start-celebrity-recognition` commande suivante lance une tâche visant à rechercher des célébrités dans le fichier vidéo spécifié stocké dans un compartiment Amazon S3.

```
aws rekognition start-celebrity-recognition \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Sortie :

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Pour plus d'informations, consultez la section Reconnaître les célébrités dans une vidéo stockée dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartCelebrityRecognitionà la section Référence des AWS CLI commandes.

`start-content-moderation`

L'exemple de code suivant montre comment utiliser`start-content-moderation`.

AWS CLI

Pour démarrer la reconnaissance d'un contenu dangereux dans une vidéo enregistrée

La `start-content-moderation` commande suivante lance une tâche visant à détecter le contenu dangereux dans le fichier vidéo spécifié stocké dans un compartiment Amazon S3.

```
aws rekognition start-content-moderation \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Sortie :

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Pour plus d'informations, consultez la section Détection des vidéos stockées non sécurisées dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartContentModerationà la section Référence des AWS CLI commandes.

`start-face-detection`

L'exemple de code suivant montre comment utiliser`start-face-detection`.

AWS CLI

Pour détecter des visages dans une vidéo

La `start-face-detection` commande suivante lance une tâche visant à détecter les visages dans le fichier vidéo spécifié stocké dans un compartiment Amazon S3.

```
aws rekognition start-face-detection
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Sortie :

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Pour plus d'informations, consultez la section Détection de visages dans une vidéo stockée dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartFaceDetectionà la section Référence des AWS CLI commandes.

`start-face-search`

L'exemple de code suivant montre comment utiliser`start-face-search`.

AWS CLI

Pour rechercher des visages dans une collection qui correspondent aux visages détectés dans une vidéo

La `start-face-search` commande suivante lance une tâche de recherche de visages dans une collection qui correspondent aux visages détectés dans le fichier vidéo spécifié dans un compartiment Amazon S3.

```
aws rekognition start-face-search \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
    --collection-id collection
```

Sortie :

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Pour plus d'informations, consultez la section Rechercher des visages dans les vidéos stockées dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartFaceSearchà la section Référence des AWS CLI commandes.

`start-label-detection`

L'exemple de code suivant montre comment utiliser`start-label-detection`.

AWS CLI

Pour détecter des objets et des scènes dans une vidéo

La `start-label-detection` commande suivante lance une tâche visant à détecter des objets et des scènes dans le fichier vidéo spécifié stocké dans un compartiment Amazon S3.

```
aws rekognition start-label-detection \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Sortie :

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Pour plus d'informations, consultez la section Détection d'étiquettes dans une vidéo du manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartLabelDetectionà la section Référence des AWS CLI commandes.

`start-person-tracking`

L'exemple de code suivant montre comment utiliser`start-person-tracking`.

AWS CLI

Pour commencer à suivre le parcours des personnes dans une vidéo stockée

La `start-person-tracking` commande suivante lance une tâche visant à suivre les chemins empruntés par les utilisateurs dans le fichier vidéo spécifié stocké dans un compartiment Amazon S3. :

```
aws rekognition start-person-tracking \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Sortie :

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Pour plus d'informations, consultez People Pathing dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartPersonTrackingà la section Référence des AWS CLI commandes.

`start-stream-processor`

L'exemple de code suivant montre comment utiliser`start-stream-processor`.

AWS CLI

Pour démarrer un processeur de flux

La `start-stream-processor` commande suivante démarre le processeur de flux vidéo spécifié.

```
aws rekognition start-stream-processor \
    --name my-stream-processor
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de vidéos en streaming dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StartStreamProcessorà la section Référence des AWS CLI commandes.

`stop-stream-processor`

L'exemple de code suivant montre comment utiliser`stop-stream-processor`.

AWS CLI

Pour arrêter un processeur de streaming en cours d'exécution

La `stop-stream-processor` commande suivante arrête le processeur de flux en cours d'exécution spécifié.

```
aws rekognition stop-stream-processor \
    --name my-stream-processor
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de vidéos en streaming dans le manuel Amazon Rekognition Developer Guide.

- Pour plus de détails sur l'API, reportez-vous StopStreamProcessorà la section Référence des AWS CLI commandes.

AWS RAM exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS RAM.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-resource-share-invitation`

L'exemple de code suivant montre comment utiliser`accept-resource-share-invitation`.

AWS CLI

Pour accepter une invitation de partage de ressources

L'`accept-resource-share-invitation`exemple suivant accepte l'invitation de partage de ressources spécifiée. Les principaux associés au compte invité peuvent immédiatement commencer à utiliser les ressources du partage.

```
aws ram accept-resource-share-invitation \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

Sortie :

```
{
    "resourceShareInvitation": {
        "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111111111111:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE",
        "resourceShareName": "MyLicenseShare",
        "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
        "senderAccountId": "111111111111",
        "receiverAccountId": "222222222222",
        "invitationTimestamp": "2021-09-22T15:07:35.620000-07:00",
        "status": "ACCEPTED"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous AcceptResourceShareInvitationà la section Référence des AWS CLI commandes.

`associate-resource-share-permission`

L'exemple de code suivant montre comment utiliser`associate-resource-share-permission`.

AWS CLI

Pour associer une autorisation gérée par la RAM à un partage de ressources

L'`associate-resource-share-permission`exemple suivant remplace l'autorisation gérée existante pour le type de ressource concerné par l'autorisation gérée spécifiée. L'accès à toutes les ressources du type de ressource concerné est régi par la nouvelle autorisation.

```
aws ram associate-resource-share-permission \
    --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite \
    --replace \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

Sortie :

```
{
    "returnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateResourceSharePermissionà la section Référence des AWS CLI commandes.

`associate-resource-share`

L'exemple de code suivant montre comment utiliser`associate-resource-share`.

AWS CLI

Exemple 1 : pour associer une ressource à un partage de ressources

L'`associate-resource-share`exemple suivant ajoute une configuration de licence au partage de ressources spécifié.

```
aws ram associate-resource-share \
    --resource-share arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
    --resource-arns arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE
```

Sortie :

```
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
            "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
            "associationType": "RESOURCE",
            "status": "ASSOCIATING",
            "external": false
        }
    ]
}
```

Exemple 2 : pour associer un principal à un partage de ressources

L'`associate-resource-share`exemple suivant accorde l'accès au partage de ressources spécifié à tous les comptes de l'unité organisationnelle spécifiée.

```
aws ram associate-resource-share \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
    --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE
```

Sortie :

```
{
    "resourceShareAssociations": [
        {
            "status": "ASSOCIATING",
            "associationType": "PRINCIPAL",
            "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
            "external": false,
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateResourceShareà la section Référence des AWS CLI commandes.

`create-resource-share`

L'exemple de code suivant montre comment utiliser`create-resource-share`.

AWS CLI

Exemple 1 : pour créer un partage de ressources

L'`create-resource-share`exemple suivant crée un partage de ressources vide portant le nom spécifié. Vous devez ajouter séparément les ressources, les principaux et les autorisations au partage.

```
aws ram create-resource-share \
    --name MyNewResourceShare
```

Sortie :

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/4476c27d-8feb-4b21-afe9-7de23EXAMPLE",
        "name": "MyNewResourceShare",
        "owningAccountId": "123456789012",
        "allowExternalPrincipals": true,
        "status": "ACTIVE",
        "creationTime": 1634586271.302,
        "lastUpdatedTime": 1634586271.302
    }
}
```

Exemple 2 : pour créer un partage de ressources avec AWS des comptes comme principaux

L'`create-resource-share`exemple suivant crée un partage de ressources et accorde l'accès au AWS compte spécifié (222222222222). Si les responsables spécifiés ne font pas partie de la même AWS organisation, les invitations sont envoyées et doivent être acceptées avant que l'accès ne soit accordé.

```
aws ram create-resource-share \
```

```
    --name MyNewResourceShare \
    --principals 222222222222
```

Exemple 3 : pour créer un partage de ressources réservé à votre AWS organisation

L'`create-resource-share`exemple suivant crée un partage de ressources limité aux comptes de l' AWS organisation dont votre compte est membre, et ajoute l'unité d'organisation spécifiée en tant que principal. Tous les comptes de cette unité d'organisation peuvent utiliser les ressources du partage de ressources.

```
aws ram create-resource-share \
    --name MyNewResourceShare \
    --no-allow-external-principals \
    --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE
```

Sortie :

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
        "name": "MyNewResourceShare",
        "owningAccountId": "123456789012",
        "allowExternalPrincipals": false,
        "status": "ACTIVE",
        "creationTime": 1634587042.49,
        "lastUpdatedTime": 1634587042.49
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateResourceShareà la section Référence des AWS CLI commandes.

`delete-resource-share`

L'exemple de code suivant montre comment utiliser`delete-resource-share`.

AWS CLI

Pour supprimer un partage de ressources

L'`delete-resource-share`exemple suivant supprime le partage de ressources spécifié.

```
aws ram delete-resource-share \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

Le résultat suivant indique un succès :

```
{
    "returnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteResourceShareà la section Référence des AWS CLI commandes.

`disassociate-resource-share-permission`

L'exemple de code suivant montre comment utiliser`disassociate-resource-share-permission`.

AWS CLI

Pour supprimer une autorisation de gestion de la RAM pour un type de ressource dans un partage de ressources

L'`disassociate-resource-share-permission`exemple suivant supprime l'autorisation de gestion de la RAM pour les bases de données Glue du partage de ressources spécifié.

```
aws ram disassociate-resource-share-permission \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
    --permission-arn arn:aws:ram::aws:permission/AWSRAMPermissionGlueDatabaseReadWrite
```

Sortie :

```
{
    "returnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateResourceSharePermissionà la section Référence des AWS CLI commandes.

`disassociate-resource-share`

L'exemple de code suivant montre comment utiliser`disassociate-resource-share`.

AWS CLI

Pour supprimer une ressource d'un partage de ressources

L'`disassociate-resource-share`exemple suivant supprime la ressource spécifiée, dans ce cas un sous-réseau VPC, du partage de ressources spécifié. Les principaux ayant accès au partage de ressources ne peuvent plus effectuer d'opérations sur cette ressource.

```
aws ram disassociate-resource-share \
    --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE
```

Sortie :

```
{
    "resourceShareAssociations": [
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
        "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE",
        "associationType": "RESOURCE",
        "status": "DISASSOCIATING",
        "external": false
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateResourceShareà la section Référence des AWS CLI commandes.

`enable-sharing-with-aws-organization`

L'exemple de code suivant montre comment utiliser`enable-sharing-with-aws-organization`.

AWS CLI

Pour permettre le partage des ressources entre les AWS Organisations

L'`enable-sharing-with-aws-organization`exemple suivant permet le partage des ressources au sein de votre organisation et de vos unités organisationnelles.

```
aws ram enable-sharing-with-aws-organization
```

La sortie suivante indique que l'opération a réussi.

```
{
    "returnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous EnableSharingWithAwsOrganizationà la section Référence des AWS CLI commandes.

`get-permission`

L'exemple de code suivant montre comment utiliser`get-permission`.

AWS CLI

Pour récupérer les détails d'une autorisation gérée par la RAM

L'`get-permission`exemple suivant affiche les détails de la version par défaut de l'autorisation de gestion de RAM spécifiée.

```
aws ram get-permission \
    --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase
```

Sortie :

```
{
    "permission": {
        "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
        "version": "2",
        "defaultVersion": true,
        "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
```

```
        "resourceType": "glue:Database",
        "permission": "{\"Effect\":\"Allow\",\"Action\":[\"glue:GetTable
\", \"glue:UpdateTable\", \"glue:DeleteTable\", \"glue:BatchDeleteTable\",
 \"glue:BatchDeleteTableVersion\", \"glue:GetTableVersion\", \"glue:GetTableVersions
\", \"glue:GetPartition\", \"glue:GetPartitions\", \"glue:BatchGetPartition\",
 \"glue:BatchCreatePartition\", \"glue:CreatePartition\", \"glue:UpdatePartition
\", \"glue:BatchDeletePartition\", \"glue:DeletePartition\", \"glue:GetTables\",
 \"glue:SearchTables\"]}",
        "creationTime": 1624912434.431,
        "lastUpdatedTime": 1624912434.431,
        "isResourceTypeDefault": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPermissionà la section Référence des AWS CLI commandes.

`get-resource-policies`

L'exemple de code suivant montre comment utiliser`get-resource-policies`.

AWS CLI

Pour obtenir les politiques relatives à une ressource

L'`get-resource-policies`exemple suivant affiche les politiques d'autorisation basées sur les ressources pour la ressource spécifiée associée à un partage de ressources.

```
aws ram get-resource-policies \
    --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE
```

Sortie :

```
{
    "policies": [
         "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"RamStatement1\",
\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[]},\"Action\":[\"ec2:RunInstances
\",\"ec2:CreateNetworkInterface\",\"ec2:DescribeSubnets\"],\"Resource\":
\"arn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1fEXAMPLE\"}]}"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourcePoliciesà la section Référence des AWS CLI commandes.

`get-resource-share-associations`

L'exemple de code suivant montre comment utiliser`get-resource-share-associations`.

AWS CLI

Exemple 1 : pour répertorier toutes les associations de ressources pour tous les types de ressources

L'`get-resource-share-associations`exemple suivant répertorie les associations de ressources pour tous les types de ressources dans tous vos partages de ressources.

```
aws ram get-resource-share-associations \
    --association-type RESOURCE
```

Sortie :

```
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE",
            "resourceShareName": "MySubnetShare",
            "associationType": "RESOURCE",
            "status": "ASSOCIATED",
            "creationTime": 1565303590.973,
            "lastUpdatedTime": 1565303591.695,
            "external": false
        },
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/8167bdfe-4480-4a01-8632-315e0EXAMPLE",
            "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
            "resourceShareName": "MyLicenseShare",
            "associationType": "RESOURCE",
            "status": "ASSOCIATED",
```

```
            "creationTime": 1632342958.457,
            "lastUpdatedTime": 1632342958.907,
            "external": false
        }
    ]
}
```

Exemple 2 : Pour répertorier les principales associations d'un partage de ressources

L'`get-resource-share-associations`exemple suivant répertorie uniquement les principales associations pour le partage de ressources spécifié uniquement.

```
aws ram get-resource-share-associations \
   --resource-share-arns arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE \
   --association-type PRINCIPAL
```

Sortie :

```
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
            "resourceShareName": "MyNewResourceShare",
            "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
            "associationType": "PRINCIPAL",
            "status": "ASSOCIATED",
            "creationTime": 1634587042.49,
            "lastUpdatedTime": 1634587044.291,
            "external": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourceShareAssociationsà la section Référence des AWS CLI commandes.

`get-resource-share-invitations`

L'exemple de code suivant montre comment utiliser`get-resource-share-invitations`.

AWS CLI

Pour répertorier vos invitations à partager des ressources

L'`get-resource-share-invitations`exemple suivant répertorie vos invitations de partage de ressources actuelles.

```
aws ram get-resource-share-invitations
```

Sortie :

```
{
    "resourceShareInvitations": [
        {
            "resourceShareInvitationArn": "arn:aws:ram:us-
west2-1:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-
e6117EXAMPLE",
            "resourceShareName": "project-resource-share",
            "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
fcb639f0-1449-4744-35bc-a983fEXAMPLE",
            "senderAccountId": "111111111111",
            "receiverAccountId": "222222222222",
            "invitationTimestamp": 1565312166.258,
            "status": "PENDING"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourceShareInvitationsà la section Référence des AWS CLI commandes.

`get-resource-shares`

L'exemple de code suivant montre comment utiliser`get-resource-shares`.

AWS CLI

Exemple 1 : Pour répertorier les partages de ressources que vous possédez et que vous partagez avec d'autres

L'`get-resource-shares`exemple suivant répertorie les partages de ressources créés et partagés avec d'autres utilisateurs.

```
aws ram get-resource-shares \
    --resource-owner SELF
```

Sortie :

```
{
    "resourceShares": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/3ab63985-99d9-1cd2-7d24-75e93EXAMPLE",
            "name": "my-resource-share",
            "owningAccountId": "123456789012",
            "allowExternalPrincipals": false,
            "status": "ACTIVE",
            "tags": [
                {
                    "key": "project",
                    "value": "lima"
                }
            ]
            "creationTime": 1565295733.282,
            "lastUpdatedTime": 1565295733.282
        },
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "name": "my-resource-share",
            "owningAccountId": "123456789012",
            "allowExternalPrincipals": true,
            "status": "ACTIVE",
            "creationTime": 1565295733.282,
            "lastUpdatedTime": 1565295733.282
        }
    ]
}
```

Exemple 2 : Pour répertorier les partages de ressources détenus par d'autres et partagés avec vous

L'`get-resource-shares`exemple suivant répertorie les partages de ressources que d'autres personnes ont créés et partagés avec vous. Dans cet exemple, il n'y en a aucune.

```
aws ram get-resource-shares \
    --resource-owner OTHER-ACCOUNTS
```

Sortie :

```
{
    "resourceShares": []
}
```

- Pour plus de détails sur l'API, reportez-vous GetResourceSharesà la section Référence des AWS CLI commandes.

`list-pending-invitation-resources`

L'exemple de code suivant montre comment utiliser`list-pending-invitation-resources`.

AWS CLI

Pour répertorier les ressources disponibles dans un partage de ressources en attente

L'`list-pending-invitation-resources`exemple suivant répertorie toutes les ressources présentes dans le partage de ressources associé à l'invitation spécifiée.

```
aws ram list-pending-invitation-resources \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:123456789012:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

Sortie :

```
{
   "resources": [
        {
            "arn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-04a555b0e6EXAMPLE",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
            "creationTime": 1634676051.269,
            "lastUpdatedTime": 1634676052.07,
            "status": "AVAILABLE",
            "type": "ec2:Subnet"
        },
```

```
        {
            "arn": "arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "creationTime": 1624912434.431,
            "lastUpdatedTime": 1624912434.431,
            "status": "AVAILABLE",
            "type": "license-manager:LicenseConfiguration"
      }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPendingInvitationResourcesà la section Référence des AWS CLI commandes.

`list-permissions`

L'exemple de code suivant montre comment utiliser`list-permissions`.

AWS CLI

Pour répertorier les autorisations gérées par la RAM disponibles

L'`list-permissions`exemple suivant répertorie toutes les autorisations gérées par la RAM disponibles uniquement pour le type de ressource de base de données AWS Glue.

```
aws ram list-permissions \
    --resource-type glue:Database
```

Sortie :

```
{
    "permissions": [
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionGlueDatabase",
            "version": "1",
            "defaultVersion": true,
            "name": "AWSRAMDefaultPermissionGlueDatabase",
            "resourceType": "glue:Database",
            "creationTime": 1592007820.935,
```

```
            "lastUpdatedTime": 1592007820.935,
            "isResourceTypeDefault": true
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueAllTablesReadWriteForDatabase",
            "version": "2",
            "defaultVersion": true,
            "name": "AWSRAMPermissionGlueAllTablesReadWriteForDatabase",
            "resourceType": "glue:Database",
            "creationTime": 1624912413.323,
            "lastUpdatedTime": 1624912413.323,
            "isResourceTypeDefault": false
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
            "version": "2",
            "defaultVersion": true,
            "name": "AWSRAMPermissionGlueDatabaseReadWrite",
            "resourceType": "glue:Database",
            "creationTime": 1624912417.4,
            "lastUpdatedTime": 1624912417.4,
            "isResourceTypeDefault": false
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
            "version": "2",
            "defaultVersion": true,
            "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
            "resourceType": "glue:Database",
            "creationTime": 1624912434.431,
            "lastUpdatedTime": 1624912434.431,
            "isResourceTypeDefault": false
        }
    ]
}
```

L'`list-permissions`exemple suivant affiche les autorisations de gestion de RAM disponibles pour tous les types de ressources.

```
aws ram list-permissions
```

Sortie :

```
{
    "permissions": [
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
            "version": "1",
            "defaultVersion": true,
            "name":
 "AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
            "resourceType": "acm-pca:CertificateAuthority",
            "creationTime": 1623264861.085,
            "lastUpdatedTime": 1623264861.085,
            "isResourceTypeDefault": false
        },
        {
            "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionAppMesh",
            "version": "1",
            "defaultVersion": true,
            "name": "AWSRAMDefaultPermissionAppMesh",
            "resourceType": "appmesh:Mesh",
            "creationTime": 1589307188.584,
            "lastUpdatedTime": 1589307188.584,
            "isResourceTypeDefault": true
        },
        ...TRUNCATED FOR BREVITY...
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
            "version": "1",
            "defaultVersion": true,
            "name":
 "AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
            "resourceType": "acm-pca:CertificateAuthority",
            "creationTime": 1623264876.75,
            "lastUpdatedTime": 1623264876.75,
            "isResourceTypeDefault": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPermissionsà la section Référence des AWS CLI commandes.

`list-principals`

L'exemple de code suivant montre comment utiliser`list-principals`.

AWS CLI

Pour répertorier les principaux ayant accès à une ressource

L'`list-principals`exemple suivant affiche une liste des principaux qui peuvent accéder aux ressources du type spécifié par le biais de n'importe quel partage de ressources.

```
aws ram list-principals \
    --resource-type ec2:Subnet
```

Sortie :

```
{
    "principals": [
        {
            "id": "arn:aws:organizations::123456789012:ou/o-gx7EXAMPLE/ou-29c5-
zEXAMPLE",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "creationTime": 1565298209.737,
            "lastUpdatedTime": 1565298211.019,
            "external": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPrincipalsà la section Référence des AWS CLI commandes.

`list-resource-share-permissions`

L'exemple de code suivant montre comment utiliser`list-resource-share-permissions`.

AWS CLI

Pour répertorier toutes les autorisations gérées par la RAM actuellement associées à un partage de ressources

L'`list-resource-share-permissions`exemple suivant répertorie toutes les autorisations gérées par la RAM associées au partage de ressources spécifié.

```
aws ram list-resource-share-permissions \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

Sortie :

```
{
    "permissions": [
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionLicenseConfiguration",
            "version": "1",
            "resourceType": "license-manager:LicenseConfiguration",
            "status": "ASSOCIATED",
            "lastUpdatedTime": 1632342984.234
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
            "version": "2",
            "resourceType": "glue:Database",
            "status": "ASSOCIATED",
            "lastUpdatedTime": 1632512462.297
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceSharePermissionsà la section Référence des AWS CLI commandes.

`list-resource-types`

L'exemple de code suivant montre comment utiliser`list-resource-types`.

AWS CLI

Pour répertorier les types de ressources pris en charge par la AWS RAM

L'`list-resource-types`exemple suivant répertorie tous les types de ressources actuellement pris en charge par la AWS RAM.

```
aws ram list-resource-types
```

Sortie :

```
{
    "resourceTypes": [
        {
            "resourceType": "route53resolver:FirewallRuleGroup",
            "serviceName": "route53resolver"
        },
        {
            "resourceType": "ec2:LocalGatewayRouteTable",
            "serviceName": "ec2"
        },
        ...OUTPUT TRUNCATED FOR BREVITY...
        {
            "resourceType": "ec2:Subnet",
            "serviceName": "ec2"
        },
        {
            "resourceType": "ec2:TransitGatewayMulticastDomain",
            "serviceName": "ec2"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceTypesà la section Référence des AWS CLI commandes.

`list-resources`

L'exemple de code suivant montre comment utiliser`list-resources`.

AWS CLI

Pour répertorier les ressources associées à un partage de ressources

L'`list-resources`exemple suivant répertorie toutes les ressources du partage de ressources spécifié qui sont du type de ressource spécifié.

```
aws ram list-resources \
    --resource-type ec2:Subnet \
    --resource-owner SELF \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

Sortie :

```
{
    "resources": [
        {
            "arn": "aarn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1f4e15235",
            "type": "ec2:Subnet",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "creationTime": 1565301545.023,
            "lastUpdatedTime": 1565301545.947
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourcesà la section Référence des AWS CLI commandes.

`promote-resource-share-created-from-policy`

L'exemple de code suivant montre comment utiliser`promote-resource-share-created-from-policy`.

AWS CLI

Pour promouvoir un partage de ressources basé sur une politique de ressources jusqu'à ce qu'il soit pleinement fonctionnel dans la RAM AWS

L'`promote-resource-share-created-from-policy`exemple suivant prend un partage de ressources que vous avez créé implicitement en attachant une politique basée sur les ressources, et le convertit pour qu'il soit pleinement fonctionnel avec la console AWS RAM et ses opérations de CLI et d'API.

```
aws ram promote-resource-share-created-from-policy \
    --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/91fa8429-2d06-4032-909a-90909EXAMPLE
```

Sortie :

```
{
    "returnValue": true
}
```

- Pour plus de détails sur l'API, reportez-vous PromoteResourceShareCreatedFromPolicyà la section Référence des AWS CLI commandes.

`reject-resource-share-invitation`

L'exemple de code suivant montre comment utiliser`reject-resource-share-invitation`.

AWS CLI

Pour rejeter une invitation de partage de ressources

L'`reject-resource-share-invitation`exemple suivant rejette l'invitation de partage de ressources spécifiée.

```
aws ram reject-resource-share-invitation \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE
```

Sortie :

```
"resourceShareInvitations": [
    {
        "resourceShareInvitationArn": "arn:aws:ram:us-west2-1:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE",
```

```
        "resourceShareName": "project-resource-share",
        "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
fcb639f0-1449-4744-35bc-a983fEXAMPLE",
        "senderAccountId": "111111111111",
        "receiverAccountId": "222222222222",
        "invitationTimestamp": 1565319592.463,
        "status": "REJECTED"
    }
]
```

- Pour plus de détails sur l'API, reportez-vous RejectResourceShareInvitationà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à un partage de ressources

L'`tag-resource`exemple suivant ajoute une clé de balise `project` et une valeur associée `lima` au partage de ressources spécifié.

```
aws ram tag-resource \
    --tags key=project,value=lima \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'un partage de ressources

L'`untag-resource`exemple suivant supprime la clé de `project` balise et la valeur associée du partage de ressources spécifié.

```
aws ram untag-resource \
    --tag-keys project \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-resource-share`

L'exemple de code suivant montre comment utiliser`update-resource-share`.

AWS CLI

Pour mettre à jour un partage de ressources

L'`update-resource-share`exemple suivant modifie le partage de ressources spécifié pour autoriser les principaux externes qui ne font pas partie d'une AWS organisation.

```
aws ram update-resource-share \
    --allow-external-principals \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

Sortie :

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
        "name": "my-resource-share",
        "owningAccountId": "123456789012",
        "allowExternalPrincipals": true,
        "status": "ACTIVE",
        "creationTime": 1565295733.282,
        "lastUpdatedTime": 1565303080.023
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateResourceShareà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Resource Explorer AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l' AWS Command Line Interface explorateur de ressources.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-default-view`

L'exemple de code suivant montre comment utiliser`associate-default-view`.

AWS CLI

Pour définir une vue de l'explorateur de ressources par défaut pour sa AWS région

L'`associate-default-view`exemple suivant définit une vue, telle que spécifiée par son ARN, comme vue par défaut pour la AWS région dans laquelle vous appelez l'opération.

```
aws resource-explorer-2 associate-default-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Sortie :

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Définition d'un affichage par défaut dans une AWS région dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous AssociateDefaultViewà la section Référence des AWS CLI commandes.

`batch-get-view`

L'exemple de code suivant montre comment utiliser`batch-get-view`.

AWS CLI

Pour récupérer des informations sur plusieurs vues de l'explorateur de ressources

L'`batch-get-view`exemple suivant affiche les détails de deux vues spécifiées par leur ARNs. Utilisez des espaces pour séparer les multiples ARNs dans le paramètre --view-arn.

```
aws resource-explorer-2 batch-get-view \
    --view-arns arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222, \
                arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Sortie :

```
{
    "Views": [
        {
            "Filters": {
                "FilterString": "service:ec2"
            },
            "IncludedProperties": [
                {
                    "Name": "tags"
                }
            ],
```

```
            "LastUpdatedAt": "2022-07-13T21:33:45.249000+00:00",
            "Owner": "123456789012",
            "Scope": "arn:aws:iam::123456789012:root",
            "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-
EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
        },
        {
            "Filters": {
                "FilterString": ""
            },
            "IncludedProperties": [
                {
                    "Name": "tags"
                }
            ],
            "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",
            "Owner": "123456789012",
            "Scope": "arn:aws:iam::123456789012:root",
            "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-
Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
        }
    ]
    "Errors": []
}
```

Pour plus d'informations sur les vues, voir À propos des vues de l'explorateur de ressources dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous BatchGetViewà la section Référence des AWS CLI commandes.

`create-index`

L'exemple de code suivant montre comment utiliser`create-index`.

AWS CLI

Pour activer l'explorateur de ressources dans une AWS région en créant un index

L'`create-index`exemple suivant crée un index local dans la AWS région dans laquelle l'opération est appelée. La AWS CLI génère automatiquement une valeur de `client-token` paramètre aléatoire et l'inclut dans l'appel à AWS si vous ne spécifiez aucune valeur.

```
aws resource-explorer-2 create-index \
    --region us-east-1
```

Sortie :

```
{
    "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222c",
    "CreatedAt": "2022-11-01T20:00:59.149Z",
    "State": "CREATING"
}
```

Après avoir créé un index local, vous pouvez le convertir en index agrégateur pour le compte en exécutant la update-index-typecommande.

Pour plus d'informations, consultez la section Activation de l'explorateur de ressources dans une AWS région pour indexer vos ressources dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous CreateIndexà la section Référence des AWS CLI commandes.

`create-view`

L'exemple de code suivant montre comment utiliser`create-view`.

AWS CLI

Exemple 1 : pour créer une vue non filtrée pour l'index dans une région AWS

L'`create-view`exemple suivant crée une vue dans la AWS région spécifiée qui renvoie tous les résultats de la région sans aucun filtrage. La vue inclut le champ facultatif Tags sur les résultats renvoyés. Comme cette vue est créée dans la région qui contient l'index agrégateur, elle peut inclure les résultats de toutes les régions du compte contenant un index Resource Explorer.

```
aws resource-explorer-2 create-view \
    --view-name My-Main-View \
    --included-properties Name=tags \
    --region us-east-1
```

Sortie :

```
{
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [
            {
                "Name": "tags"
            }
        ],
        "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
    }
}
```

Exemple 2 : pour créer une vue qui renvoie uniquement les ressources associées à Amazon EC2

Ce qui suit `create-view` crée une vue dans AWS Region `us-east-1` qui renvoie uniquement les ressources de la Region associées au EC2 service Amazon. La vue inclut le `Tags` champ facultatif sur les résultats renvoyés. Comme cette vue est créée dans la région qui contient l'index agrégateur, elle peut inclure les résultats de toutes les régions du compte contenant un index Resource Explorer.

```
aws resource-explorer-2 create-view \
    --view-name My-EC2-Only-View \
    --included-properties Name=tags \
    --filters FilterString="service:ec2" \
    --region us-east-1
```

Sortie :

```
{
    "View": {
        "Filters": {
            "FilterString": "service:ec2"
        },
        "IncludedProperties": [
```

```
            {
                "Name":"tags"
            }
        ],
        "LastUpdatedAt": "2022-07-13T21:35:09.059Z",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
}
```

Pour plus d'informations, consultez la section Création de vues pour la recherche dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous CreateViewà la section Référence des AWS CLI commandes.

`delete-index`

L'exemple de code suivant montre comment utiliser`delete-index`.

AWS CLI

Pour désactiver l'explorateur de ressources dans une AWS région en supprimant son index

L'`delete-index`exemple suivant supprime l'index Resource Explorer spécifié dans la AWS région dans laquelle vous faites la demande.

```
aws resource-explorer-2 delete-index \
    --arn arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222 \
    --region us-west-2
```

Sortie :

```
{
    "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222",
    "State": "DELETING"
}
```

Pour plus d'informations sur la suppression d'un index, consultez la section Désactivation de l'explorateur de AWS ressources dans une AWS région dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous DeleteIndexà la section Référence des AWS CLI commandes.

`delete-view`

L'exemple de code suivant montre comment utiliser`delete-view`.

AWS CLI

Pour supprimer une vue de l'explorateur de ressources

L'`delete-view`exemple suivant supprime une vue spécifiée par son ARN.

```
aws resource-explorer-2 delete-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Sortie :

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Suppression de vues dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous DeleteViewà la section Référence des AWS CLI commandes.

`disassociate-default-view`

L'exemple de code suivant montre comment utiliser`disassociate-default-view`.

AWS CLI

Pour supprimer la vue de l'explorateur de ressources par défaut pour une AWS région

Ce qui suit `disassociate-default-view` supprime la vue par défaut de l'explorateur de ressources pour la AWS région dans laquelle vous appelez l'opération. Après avoir effectué cette opération, toutes les opérations de recherche dans la région doivent spécifier explicitement une vue, sinon l'opération échoue.

```
aws resource-explorer-2 disassociate-default-view
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition d'un affichage par défaut dans une AWS région dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous DisassociateDefaultViewà la section Référence des AWS CLI commandes.

`get-default-view`

L'exemple de code suivant montre comment utiliser`get-default-view`.

AWS CLI

Pour récupérer la vue Resource Explorer, qui est la vue par défaut pour sa AWS région

L'`get-default-view`exemple suivant récupère l'ARN de la vue par défaut pour la AWS région dans laquelle vous appelez l'opération.

```
aws resource-explorer-2 get-default-view
```

Sortie :

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Définition d'un affichage par défaut dans une AWS région dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous GetDefaultViewà la section Référence des AWS CLI commandes.

`get-index`

L'exemple de code suivant montre comment utiliser`get-index`.

AWS CLI

Exemple 1 : pour récupérer les détails d'un index agrégateur Resource Explorer

L'`get-index`exemple suivant affiche les détails de l'index Resource Explorer dans la AWS région spécifiée. Comme la région spécifiée contient l'index agrégateur du compte, la sortie répertorie les régions qui répliquent les données dans l'index de cette région.

```
aws resource-explorer-2 get-index \
    --region us-east-1
```

Sortie :

```
{
    "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE11111",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingFrom": [
        "ap-south-1",
        "us-west-2"
    ],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "AGGREGATOR"
}
```

Exemple 2 : pour récupérer les détails d'un index local de Resource Explorer

L'`get-index`exemple suivant affiche les détails de l'index Resource Explorer dans la AWS région spécifiée. Comme la région spécifiée contient un index local, la sortie répertorie la région vers laquelle elle réplique les données de l'index de cette région.

```
aws resource-explorer-2 get-index \
    --region us-west-2
```

Sortie :

```
{
    "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingTo": [
        "us-west-2"
    ],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "LOCAL"
}
```

Pour plus d'informations sur les index, consultez la section Vérification des AWS régions dans lesquelles l'explorateur de ressources est activé dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous GetIndexà la section Référence des AWS CLI commandes.

`get-view`

L'exemple de code suivant montre comment utiliser`get-view`.

AWS CLI

Pour récupérer les informations relatives à une vue de l'explorateur de ressources

L'`get-view`exemple suivant affiche les détails d'une vue spécifiée par son ARN.

```
aws resource-explorer-2 get-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Sortie :

```
{
  "Tags" : {},
  "View" : {
        "Filters" : {
            "FilterString" : "service:ec2"
        },
```

```
        "IncludedProperties" : [
            {
                "Name" : "tags"
            }
        ],
        "LastUpdatedAt" : "2022-07-13T21:33:45.249Z",
        "Owner" : "123456789012",
        "Scope" : "arn:aws:iam::123456789012:root",
        "ViewArn" : "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }
}
```

Pour plus d'informations sur les vues, voir À propos des vues de l'explorateur de ressources dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous GetViewà la section Référence des AWS CLI commandes.

`list-indexes`

L'exemple de code suivant montre comment utiliser`list-indexes`.

AWS CLI

Pour répertorier les AWS régions dans lesquelles Resource Explorer possède des index

L'`list-indexes`exemple suivant répertorie les index de toutes les régions où Resource Explorer possède un index. La réponse indique le type de chaque index, sa AWS région et son ARN.

```
aws resource-explorer-2 list-indexes
```

Sortie :

```
{
    "Indexes": [
        {
            "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
            "Region": "us-west-2",
            "Type": "AGGREGATOR"
        },
        {
```

```
            "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
            "Region": "us-east-1",
            "Type": "LOCAL"
        },
        {
            "Arn": "arn:aws:resource-explorer-2:us-east-2:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333",
            "Region": "us-east-2",
            "Type": "LOCAL"
        },
        {
            "Arn": "arn:aws:resource-explorer-2:us-west-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE44444",
            "Region": "us-west-1",
            "Type": "LOCAL"
        }
    ]
}
```

Pour plus d'informations sur les index, consultez la section Vérification des AWS régions dans lesquelles l'explorateur de ressources est activé dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous ListIndexesà la section Référence des AWS CLI commandes.

`list-supported-resource-types`

L'exemple de code suivant montre comment utiliser`list-supported-resource-types`.

AWS CLI

Pour répertorier les AWS régions dans lesquelles Resource Explorer possède des index

L'`list-supported-resource-types`exemple suivant répertorie tous les types de ressources actuellement pris en charge par & AREXlong ;. L'exemple de réponse inclut une `NextToken` valeur qui indique qu'il y a davantage de sorties disponibles à récupérer avec des appels supplémentaires.

```
aws resource-explorer-2 list-supported-resource-types \
    --max-items 10
```

Sortie :


```
{
    "ResourceTypes": [
        {
            "ResourceType": "cloudfront:cache-policy",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:distribution",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:function",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:origin-access-identity",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:origin-request-policy",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:realtime-log-config",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:response-headers-policy",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudwatch:alarm",
            "Service": "cloudwatch"
        },
        {
            "ResourceType": "cloudwatch:dashboard",
            "Service": "cloudwatch"
        },
        {
            "ResourceType": "cloudwatch:insight-rule",
            "Service": "cloudwatch"
        }
```

```
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

Pour obtenir la partie suivante de la sortie, appelez à nouveau l'opération et transmettez la valeur de `NextToken` réponse de l'appel précédent comme valeur pour`--starting-token`. Répétez jusqu'à ce qu'`NextToken`il soit absent de la réponse.

```
aws resource-explorer-2 list-supported-resource-types \
    --max-items 10  \
    --starting-
token eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=
```

Sortie :

```
{
    "ResourceTypes": [
        {
            "ResourceType": "cloudwatch:metric-stream",
            "Service": "cloudwatch"
        },
        {
            "ResourceType": "dynamodb:table",
            "Service": "dynamodb"
        },
        {
            "ResourceType": "ec2:capacity-reservation",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:capacity-reservation-fleet",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:client-vpn-endpoint",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:customer-gateway",
            "Service": "ec2"
        },
        {
```

```
            "ResourceType": "ec2:dedicated-host",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:dhcp-options",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:egress-only-internet-gateway",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:elastic-gpu",
            "Service": "ec2"
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyMH0="
}
```

Pour plus d'informations sur les index, consultez la section Vérification des AWS régions dans lesquelles l'explorateur de ressources est activé dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous ListSupportedResourceTypesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises associées à une vue ou à un index de l'Explorateur de ressources

L'`list-tags-for-resource`exemple suivant répertorie les paires clé et valeur de balise associées à afficher avec l'ARN spécifié. Vous devez appeler l'opération depuis la AWS région qui contient la ressource.

```
aws resource-explorer-2 list-tags-for-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Sortie :

```
{
    "Tags": {
        "application": "MainCorpApp",
        "department": "1234"
    }
}
```

Pour plus d'informations sur le balisage des vues, consultez la section Balisage des vues pour le contrôle d'accès dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-views`

L'exemple de code suivant montre comment utiliser`list-views`.

AWS CLI

Pour répertorier les vues de l'explorateur de ressources disponibles dans une AWS région

L'`list-views`exemple suivant répertorie toutes les vues disponibles dans la région dans laquelle vous appelez l'opération.

```
aws resource-explorer-2 list-views
```

Sortie :

```
{
    "Views": [
        "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
        "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Default-All-
Resources-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
        "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Production-Only-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333"
    ]
}
```

Pour plus d'informations sur les vues, voir À propos des vues de l'explorateur de ressources dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous ListViewsà la section Référence des AWS CLI commandes.

`search`

L'exemple de code suivant montre comment utiliser`search`.

AWS CLI

Exemple 1 : Pour effectuer une recherche à l'aide de la vue par défaut

L'`search`exemple suivant affiche toutes les ressources spécifiées associées au service. La recherche utilise la vue par défaut pour la région. L'exemple de réponse inclut une `NextToken` valeur qui indique qu'il y a davantage de sorties disponibles à récupérer avec des appels supplémentaires.

```
aws resource-explorer-2 search \
    --query-string "service:iam"
```

Sortie :

```
{
    "Count": {
        "Complete": true,
        "TotalResources": 55
    },
    "NextToken":
 "AG9VOEF1KLEXAMPLEOhJHVwo5chEXAMPLER5XiEpNrgsEXAMPLE...b0CmOFOryHEXAMPLE",
    "Resources": [{
        "Arn": "arn:aws:iam::123456789012:policy/service-role/Some-Policy-For-A-
Service-Role",
        "LastReportedAt": "2022-07-21T12:34:42Z",
        "OwningAccountId": "123456789012",
        "Properties": [],
        "Region": "global",
        "ResourceType": "iam:policy",
        "Service": "iam"
    }, {
```

```
        "Arn": "arn:aws:iam::123456789012:policy/service-role/Another-Policy-For-A-
Service-Role",
        "LastReportedAt": "2022-07-21T12:34:42Z",
        "OwningAccountId": "123456789012",
        "Properties": [],
        "Region": "global",
        "ResourceType": "iam:policy",
        "Service": "iam"
    }, {
       ... TRUNCATED FOR BREVITY ...
    }],
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/my-default-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Exemple 2 : Pour effectuer une recherche à l'aide d'une vue spécifiée

L'`search`exemple de recherche suivant affiche toutes les ressources (« * ») de la AWS région spécifiée qui sont visibles dans la vue spécifiée. Les résultats incluent uniquement les ressources associées à Amazon en EC2 raison des filtres attachés à la vue.

```
aws resource-explorer-2 search \
    -- query-string "*" \
    -- view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-view/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

Sortie :

```
HTTP/1.1 200 OK
Date: Tue, 01 Nov 2022 20:00:59 GMT
Content-Type: application/json
Content-Length: <PayloadSizeBytes>

    {
        "Count": {
            "Complete": true,
            "TotalResources": 67
        },
        "Resources": [{
            "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/acl-1a2b3c4d",
            "LastReportedAt": "2022-07-21T18:52:02Z",
            "OwningAccountId": "123456789012",
            "Properties": [{
```

```
                "Data": [{
                    "Key": "Department",
                    "Value": "AppDevelopment"
                }, {
                    "Key": "Environment",
                    "Value": "Production"
                }],
                "LastReportedAt": "2021-11-15T14:48:29Z",
                "Name": "tags"
            }],
            "Region": "us-east-1",
            "ResourceType": "ec2:network-acl",
            "Service": "ec2"
        }, {
            "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-1a2b3c4d",
            "LastReportedAt": "2022-07-21T21:22:23Z",
            "OwningAccountId": "123456789012",
            "Properties": [{
                "Data": [{
                    "Key": "Department",
                    "Value": "AppDevelopment"
                }, {
                    "Key": "Environment",
                    "Value": "Production"
                }],
                "LastReportedAt": "2021-07-29T19:02:39Z",
                "Name": "tags"
            }],
            "Region": "us-east-1",
            "ResourceType": "ec2:subnet",
            "Service": "ec2"
        }, {
            "Arn": "arn:aws:ec2:us-east-1:123456789012:dhcp-options/dopt-1a2b3c4d",
            "LastReportedAt": "2022-07-21T06:08:53Z",
            "OwningAccountId": "123456789012",
            "Properties": [{
                "Data": [{
                    "Key": "Department",
                    "Value": "AppDevelopment"
                }, {
                    "Key": "Environment",
                    "Value": "Production"
                }],
                "LastReportedAt": "2021-11-15T15:11:05Z",
```

```
                "Name": "tags"
            }],
            "Region": "us-east-1",
            "ResourceType": "ec2:dhcpoptions",
            "Service": "ec2"
        }, {
            ... TRUNCATED FOR BREVITY ...
        }],
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
```

Pour plus d'informations, consultez la section Utilisation de l'explorateur de AWS ressources pour rechercher des ressources dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, consultez la section Rechercher dans AWS CLI la référence des commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour baliser une vue de l'explorateur de ressources

L'`tag-resource`exemple suivant ajoute la clé de balise « environment » avec la valeur « production » à la vue avec l'ARN spécifié.

```
aws resource-explorer-2 tag-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
    --tags environment=production
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des vues pour le contrôle d'accès dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une balise d'une vue de l'explorateur de ressources

L'`untag-resource`exemple suivant supprime toute balise portant le nom de clé « environment » de la vue avec l'ARN spécifié.

```
aws resource-explorer-2 untag-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
    --tag-keys environment
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des vues pour le contrôle d'accès dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-index-type`

L'exemple de code suivant montre comment utiliser`update-index-type`.

AWS CLI

Pour modifier le type d'un index de l'explorateur de ressources

L'`update-index-type`exemple suivant convertit l'index spécifié de type `local` en type `aggregator` pour activer la possibilité de rechercher des ressources dans toutes les AWS régions du compte. Vous devez envoyer la demande à la AWS région qui contient l'index que vous souhaitez mettre à jour.

```
aws resource-explorer-2 update-index-type \
    --arn arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE11111 \
```

```
    --type aggregator \
    --region us-east-1
```

Sortie :

```
{
    "Arn":"arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE11111",
    "LastUpdatedAt":"2022-07-13T18:41:58.799Z",
    "State":"updating",
    "Type":"aggregator"
}
```

Pour plus d'informations sur la modification du type d'un index, voir Activer la recherche interrégionale en créant un index agrégateur dans le guide de l'utilisateur de AWS Resource Explorer.

- Pour plus de détails sur l'API, reportez-vous UpdateIndexTypeà la section Référence des AWS CLI commandes.

`update-view`

L'exemple de code suivant montre comment utiliser`update-view`.

AWS CLI

Exemple 1 : pour mettre à jour le IncludedProperties champ d'une vue de l'explorateur de ressources

L'`update-view`exemple suivant met à jour la vue spécifiée en l'ajoutant `` `tags` `` à la vue facultative`` `IncludedProperties` ``. Après avoir exécuté cette opération, les opérations de recherche qui utilisent cette vue incluent des informations sur les balises associées aux ressources qui apparaissent dans les résultats.

```
aws resource-explorer-2 update-view \
    --included-properties Name=tags \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

Sortie :

```
{
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [
            {
                "Name": "tags"
            }
        ],
        "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
    }
}
```

Exemple 2 : pour mettre à jour les filtres associés à une vue

L'`update-view`exemple suivant met à jour la vue spécifiée pour utiliser un filtre qui limite les résultats aux seuls types de ressources associés au EC2 service Amazon.

```
aws resource-explorer-2 update-view \
    --filters FilterString="service:ec2" \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

Sortie :

```
{
    "View": {
    "Filters": {
        "FilterString": "service:ec2"
    },
    "IncludedProperties": [],
    "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
```

```
}
```

Pour plus d'informations sur les vues, voir À propos des vues de l'explorateur de ressources dans le guide de l'utilisateur de l'explorateur de AWS ressources.

- Pour plus de détails sur l'API, reportez-vous UpdateViewà la section Référence des AWS CLI commandes.

Exemples de Resource Groups utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Resource Groups.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Exemple 1 : pour créer un groupe de ressources basé sur des balises

L'`create-group`exemple suivant crée un groupe de ressources basé sur des balises d' EC2 instances Amazon dans la région actuelle. Il est basé sur une requête pour les ressources étiquetées avec la clé `Name` et la valeur`WebServers`. Le nom du groupe est`tbq-WebServer`. La requête se trouve dans un fichier JSON distinct qui est transmis à la commande.

```
aws resource-groups create-group \
```

```
    --name tbq-WebServer \
    --resource-query file://query.json
```

Contenu de `query.json` :

```
{
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
}
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    },
    "ResourceQuery": {
        "Type": "TAG_FILTERS_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
    }
}
```

Exemple 2 : pour créer un groupe de ressources CloudFormation basé sur une pile

L'`create-group`exemple suivant crée un groupe de ressources AWS CloudFormation basé sur une pile nommé. `sampleCFNstackgroup` La requête inclut toutes les ressources de la CloudFormation pile spécifiée qui sont prises en charge par AWS Resource Groups.

```
aws resource-groups create-group \
    --name cbq-CFNstackgroup \
    --resource-query file://query.json
```

Contenu de `query.json` :

```
{
    "Type": "CLOUDFORMATION_STACK_1_0",
```

```
    "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
}
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNstackgroup",
        "Name": "cbq-CFNstackgroup"
    },
    "ResourceQuery": {
        "Type": "CLOUDFORMATION_STACK_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-east-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"}'
    }
}
```

Pour plus d'informations, consultez Create Groups dans le guide de l'utilisateur de AWS Resource Groups.

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour mettre à jour la description d'un groupe de ressources

L'`delete-group`exemple suivant met à jour le groupe de ressources spécifié.

```
aws resource-groups delete-group \
    --group-name tbq-WebServer
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    }
}
```

Pour plus d'informations, consultez la section Delete Groups dans le guide de l'utilisateur de AWS Resource Groups.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`get-group-query`

L'exemple de code suivant montre comment utiliser`get-group-query`.

AWS CLI

Pour associer la requête à un groupe de ressources

L'`get-group-query`exemple suivant montre une requête attachée au groupe de ressources spécifié.

```
aws resource-groups get-group-query \
    --group-name tbq-WebServer
```

Sortie :

```
{
    "GroupQuery": {
        "GroupName": "tbq-WebServer",
        "ResourceQuery": {
            "Type": "TAG_FILTERS_1_0",
            "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters
\":[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetGroupQueryà la section Référence des AWS CLI commandes.

`get-group`

L'exemple de code suivant montre comment utiliser`get-group`.

AWS CLI

Pour obtenir des informations sur un groupe de ressources

L'`get-group`exemple suivant affiche des détails sur le groupe de ressources spécifié. Pour associer la requête au groupe, utilisez`get-group-query`.

```
aws resource-groups get-group \
    --group-name tbq-WebServer
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer",
        "Description": "A tag-based query resource group of WebServers."
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetGroupà la section Référence des AWS CLI commandes.

`get-tags`

L'exemple de code suivant montre comment utiliser`get-tags`.

AWS CLI

Pour récupérer les balises associées à un groupe de ressources

L'`get-tags`exemple suivant affiche les paires clé et valeur de balise associées au groupe de ressources spécifié (le groupe lui-même, pas ses membres).

```
aws resource-groups get-tags \
    --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

Sortie :

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Tags": {
        "QueryType": "tags",
        "QueryResources": "ec2-instances"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetTagsà la section Référence des AWS CLI commandes.

`list-group-resources`

L'exemple de code suivant montre comment utiliser`list-group-resources`.

AWS CLI

Pour répertorier toutes les ressources d'un groupe de ressources

Exemple 1 : L'`list-resource-groups`exemple suivant répertorie toutes les ressources qui font partie du groupe de ressources spécifié.

```
aws resource-groups list-group-resources \
    --group-name tbq-WebServer
```

Sortie :

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
```

```
}
```

Exemple 2 : L'exemple suivant répertorie toutes les ressources du groupe qui ont également un « type de ressource » du type « AWS : : : EC2 :Instance » . :

aws resource-groups list-group-resources --group-name tbq- WebServer --filters Name=Type de ressource, Valeurs= : : :Instance AWS EC2

- Pour plus de détails sur l'API, reportez-vous ListGroupResourcesà la section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour répertorier les groupes de ressources disponibles

L'`list-groups`exemple suivant affiche une liste de tous les groupes de ressources.

```
aws resource-groups list-groups
```

Sortie :

```
{
    "GroupIdentifiers": [
        {
            "GroupName": "tbq-WebServer",
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer3"
        },
        {
            "GroupName": "cbq-CFNStackQuery",
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNStackQuery"
        }
    ],
    "Groups": [
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
```

```
            "Name": "tbq-WebServer"
        },
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNStackQuery",
            "Name": "cbq-CFNStackQuery"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGroupsà la section Référence des AWS CLI commandes.

`list-resource-groups`

L'exemple de code suivant montre comment utiliser`list-resource-groups`.

AWS CLI

Pour répertorier toutes les ressources d'un groupe de ressources

L'`list-resource-groups`exemple suivant répertorie toutes les ressources qui font partie du groupe de ressources spécifié.

```
aws resource-groups list-group-resources \
    --group-name tbq-WebServer
```

Sortie :

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceGroupsà la section Référence des AWS CLI commandes.

`put-group-configuration`

L'exemple de code suivant montre comment utiliser`put-group-configuration`.

AWS CLI

Pour associer une configuration de service à un groupe de ressources

Exemple 1 : L'`put-group-configuration`exemple suivant indique que le groupe de ressources doit contenir uniquement les réservations de EC2 capacité Amazon pour les instances appartenant à la `M5` famille `C5` or.

```
aws resource-groups put-group-configuration \
    --group MyTestGroup \
    --configuration file://config.json
```

Contenu de `config.json` :

```
[
    {
        "Type": "AWS::EC2::HostManagement",
        "Parameters": [
            {
                "Name": "allowed-host-families",
                "Values": [ "c5", "m5" ]
            },
                {
                    "Name": "any-host-based-license-configuration",
                    "Values": [ "true" ]
                }
        ]
    },
    {
        "Type": "AWS::ResourceGroups::Generic",
        "Parameters": [
            {
                "Name": "allowed-resource-types",
                "Values": [ "AWS::EC2::Host" ]
            },
            {
                "Name": "deletion-protection",
                "Values": [ "UNLESS_EMPTY" ]
            }
```

```
        ]
    }
]
```

Cette commande ne produit aucune sortie en cas de réussite.

Pour plus d'informations, consultez la section Configurations de service pour les groupes de ressources dans le Guide de référence de l'API Resource Groups.

- Pour plus de détails sur l'API, reportez-vous PutGroupConfigurationà la section Référence des AWS CLI commandes.

`search-resources`

L'exemple de code suivant montre comment utiliser`search-resources`.

AWS CLI

Pour rechercher des ressources correspondant à une requête

L'`search-resources`exemple suivant extrait une liste de toutes les AWS ressources qui correspondent à la requête spécifiée.

```
aws resource-groups search-resources \
    --resource-query file://query.json
```

Contenu de `query.json` :

```
{
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Patch Group\", \"Values\":[\"Dev\"]}]}"
}
```

Sortie :

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a23bc45d67890ef",
```

```
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous SearchResourcesà la section Référence des AWS CLI commandes.

`tag`

L'exemple de code suivant montre comment utiliser`tag`.

AWS CLI

Pour associer une étiquette à un groupe de ressources

L'`tag`exemple suivant attache les paires clé et valeur de balise spécifiées au groupe de ressources spécifié (le groupe lui-même, pas ses membres).

```
aws resource-groups tag \
    --tags QueryType=tags,QueryResources=ec2-instances \
    --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer
```

Sortie :

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
    "Tags": {
        "QueryType": "tags",
        "QueryResources": "ec2-instances"
    }
}
```

Pour plus d'informations, consultez la section Gérer les balises dans le guide de l'utilisateur de AWS Resource Groups.

- Pour plus de détails sur l'API, voir Tag in AWS CLI Command Reference.

`untag`

L'exemple de code suivant montre comment utiliser`untag`.

AWS CLI

Pour supprimer des balises d'un groupe de ressources

L'`untags`exemple suivant supprime toute balise avec la clé spécifiée du groupe de ressources lui-même, et non de ses membres.

```
aws resource-groups untag \
    --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \
    --keys QueryType
```

Sortie :

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Keys": [
        "QueryType"
    ]
}
```

Pour plus d'informations, consultez la section Gérer les balises dans le guide de l'utilisateur de AWS Resource Groups.

- Pour plus de détails sur l'API, voir Untag dans la référence des AWS CLI commandes.

`update-group-query`

L'exemple de code suivant montre comment utiliser`update-group-query`.

AWS CLI

Exemple 1 : pour mettre à jour la requête pour un groupe de ressources basé sur des balises

L'`update-group-query`exemple suivant met à jour la requête attachée au groupe de ressources basé sur des balises spécifié.

```
aws resource-groups update-group-query \
    --group-name tbq-WebServer \
    --resource-query '{"Type":"TAG_FILTERS_1_0", "Query":"{\"ResourceTypeFilters\":
[\"AWS::EC2::Instance\"],\"TagFilters\":[{\"Key\":\"Name\", \"Values\":[\"WebServers
\"]}]}"}'
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-east-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    },
    "ResourceQuery": {
        "Type": "TAG_FILTERS_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
    }
}
```

Pour plus d'informations, consultez la section Update Groups dans le guide de l'utilisateur de AWS Resource Groups.

Exemple 2 : pour mettre à jour la requête pour un groupe de ressources CloudFormation basé sur une pile

L'`update-group-query`exemple suivant met à jour la requête attachée au groupe de ressources AWS CloudFormation basé sur une pile spécifié.

```
aws resource-groups update-group-query \
    --group-name cbq-CFNstackgroup \
    --resource-query '{"Type": "CLOUDFORMATION_STACK_1_0", "Query":
 "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier\":
\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"}'
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNstackgroup",
        "Name": "cbq-CFNstackgroup"
    },
    "ResourceQuery": {
        "Type": "CLOUDFORMATION_STACK_1_0",
```

```
        "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
    }
}
```

Pour plus d'informations, consultez la section Update Groups dans le guide de l'utilisateur de AWS Resource Groups.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupQueryà la section Référence des AWS CLI commandes.

`update-group`

L'exemple de code suivant montre comment utiliser`update-group`.

AWS CLI

Pour mettre à jour la description d'un groupe de ressources

L'`update-group`exemple suivant met à jour la description du groupe de ressources spécifié.

```
aws resource-groups update-group \
    --group-name tbq-WebServer \
    --description "Resource group for all web server resources."
```

Sortie :

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
        "Description": "Resource group for all web server resources."
    }
}
```

Pour plus d'informations, consultez la section Update Groups dans le guide de l'utilisateur de AWS Resource Groups.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupà la section Référence des AWS CLI commandes.

Exemples d'API de balisage Resource Groups utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l'API AWS Command Line Interface with Resource Groups Tagging.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-resources`

L'exemple de code suivant montre comment utiliser`get-resources`.

AWS CLI

Pour obtenir une liste de ressources balisées

L'`get-resources`exemple suivant affiche une liste des ressources du compte qui sont étiquetées avec le nom et la valeur de clé spécifiés.

```
aws resourcegroupstaggingapi get-resources \
    --tag-filters Key=Environment,Values=Production \
    --tags-per-page 100
```

Sortie :

```
{
    "ResourceTagMappingList": [
        {
            "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-7sbz2Kz0",
```

```
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez le GetResourcesmanuel Resource Groups Tagging API Reference.

- Pour plus de détails sur l'API, reportez-vous GetResourcesà la section Référence des AWS CLI commandes.

`get-tag-keys`

L'exemple de code suivant montre comment utiliser`get-tag-keys`.

AWS CLI

Pour obtenir une liste de toutes les clés de tag

L'`get-tag-keys`exemple suivant extrait la liste de tous les noms de clés de balise utilisés par les ressources du compte.

```
aws resourcegroupstaggingapi get-tag-keys
```

Sortie :

```
{
    "TagKeys": [
        "Environment",
        "CostCenter",
        "Department"
    ]
}
```

Pour plus d'informations, consultez le GetTagKeysmanuel Resource Groups Tagging API Reference.

- Pour plus de détails sur l'API, reportez-vous GetTagKeysà la section Référence des AWS CLI commandes.

`get-tag-values`

L'exemple de code suivant montre comment utiliser`get-tag-values`.

AWS CLI

Pour obtenir une liste de toutes les valeurs des balises

L'`get-tag-values`exemple suivant affiche toutes les valeurs utilisées pour la clé spécifiée pour toutes les ressources du

```
aws resourcegroupstaggingapi get-tag-values \
    --key=Environment
```

Sortie :

```
{
    "TagValues": [
        "Alpha",
        "Gamma",
        "Production"
    ]
}
```

Pour plus d'informations, consultez le GetTagValuesmanuel Resource Groups Tagging API Reference.

- Pour plus de détails sur l'API, reportez-vous GetTagValuesà la section Référence des AWS CLI commandes.

`tag-resources`

L'exemple de code suivant montre comment utiliser`tag-resources`.

AWS CLI

Pour associer un tag à une ressource

L'`tag-resources`exemple suivant balise la ressource spécifiée avec un nom et une valeur de clé.

```
aws resourcegroupstaggingapi tag-resources \
    --resource-arn-list arn:aws:s3:::MyProductionBucket \
    --tags Environment=Production,CostCenter=1234
```

Sortie :

```
{
    "FailedResourcesMap": {}
}
```

Pour plus d'informations, consultez le TagResourcesmanuel Resource Groups Tagging API Reference.

- Pour plus de détails sur l'API, reportez-vous TagResourcesà la section Référence des AWS CLI commandes.

`untag-resources`

L'exemple de code suivant montre comment utiliser`untag-resources`.

AWS CLI

Pour supprimer un tag d'une ressource

L'`untag-resources`exemple suivant supprime les clés de balise spécifiées et toutes les valeurs associées de la ressource spécifiée.

```
aws resourcegroupstaggingapi untag-resources \
    --resource-arn-list arn:aws:s3:::amzn-s3-demo-bucket \
    --tag-keys Environment CostCenter
```

Sortie :

```
{
    "FailedResourcesMap": {}
}
```

Pour plus d'informations, consultez le UntagResourcesmanuel Resource Groups Tagging API Reference.

- Pour plus de détails sur l'API, reportez-vous UntagResourcesà la section Référence des AWS CLI commandes.

AWS RoboMaker exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS RoboMaker.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-describe-simulation-job`

L'exemple de code suivant montre comment utiliser`batch-describe-simulation-job`.

AWS CLI

Pour décrire des tâches de simulation par lots

L'`batch-describe-simulation-job`exemple suivant permet de récupérer les détails des trois tâches de simulation spécifiées.

Commande :

```
aws robomaker batch-describe-simulation-job \
--job arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-p0cpdrrwng2n arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tglmblgw
```

Sortie :

```
{
    "jobs": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x",
            "status": "Completed",
            "lastUpdatedAt": 1548959178.0,
            "failureBehavior": "Continue",
            "clientRequestToken": "6020408e-b05c-4310-9f13-4ed71c5221ed",
            "outputLocation": {
                "s3Bucket": "awsrobomakerobjecttracker-111111111-
bundlesbucket-2lk584kiq1oa",
                "s3Prefix": "output"
            },
            "maxJobDurationInSeconds": 3600,
            "simulationTimeMillis": 0,
            "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerObjectTracker-154895-SimulationJobRole-14D5ASA7PQE3A",
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "object_tracker_simulation",
                        "launchFile": "local_training.launch",
                        "environmentVariables": {
                            "MARKOV_PRESET_FILE": "object_tracker.py",
                            "MODEL_S3_BUCKET": "awsrobomakerobjecttracker-111111111-
bundlesbucket-2lk584kiq1oa",
                            "MODEL_S3_PREFIX": "model-store",
                            "ROS_AWS_REGION": "us-west-2"
                        }
                    }
                }
            ],
            "tags": {},
            "vpcConfig": {
                "subnets": [
                    "subnet-716dd52a",
                    "subnet-43c22325",
```

```
                    "subnet-3f526976"
                ],
                "securityGroups": [
                    "sg-3fb40545"
                ],
                "vpcId": "vpc-99895eff",
                "assignPublicIp": true
            }
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
p0cpdrrwng2n",
            "status": "Completed",
            "lastUpdatedAt": 1548168817.0,
            "failureBehavior": "Continue",
            "clientRequestToken": "e4a23e75-f9a7-411d-835f-21881c82c58b",
            "outputLocation": {
                "s3Bucket": "awsrobomakercloudwatch-111111111111-
bundlesbucket-14e5s9jvwtmv7",
                "s3Prefix": "output"
            },
            "maxJobDurationInSeconds": 3600,
            "simulationTimeMillis": 0,
            "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G0OBWTQ8YBG6",
            "robotApplications": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerCloudWatch-1547663411642_NZbpqEJ3T/1547663517377",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "cloudwatch_robot",
                        "launchFile": "await_commands.launch",
                        "environmentVariables": {
                            "LAUNCH_ID": "1548168752173",
                            "ROS_AWS_REGION": "us-west-2"
                        }
                    }
                }
            ],
            "simulationApplications": [
                {
```

```
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "cloudwatch_simulation",
                        "launchFile": "bookstore_turtlebot_navigation.launch",
                        "environmentVariables": {
                            "LAUNCH_ID": "1548168752173",
                            "ROS_AWS_REGION": "us-west-2",
                            "TURTLEBOT3_MODEL": "waffle_pi"
                        }
                    }
                }
            ],
            "tags": {},
            "vpcConfig": {
                "subnets": [
                    "subnet-716dd52a",
                    "subnet-43c22325",
                    "subnet-3f526976"
                ],
                "securityGroups": [
                    "sg-3fb40545"
                ],
                "vpcId": "vpc-99895eff",
                "assignPublicIp": true
            }
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tglmblgw",
            "status": "Canceled",
            "lastUpdatedAt": 1546543442.0,
            "failureBehavior": "Fail",
            "clientRequestToken": "d796bbb4-2a2c-1abc-f2a9-0d9e547d853f",
            "outputLocation": {
                "s3Bucket": "sample-bucket",
                "s3Prefix": "SimulationLog_115490482698"
            },
            "maxJobDurationInSeconds": 28800,
            "simulationTimeMillis": 0,
            "iamRole": "arn:aws:iam::111111111111:role/RoboMakerSampleTheFirst",
            "robotApplications": [
```

```
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "hello_world_robot",
                        "launchFile": "rotate.launch"
                    }
                }
            ],
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "hello_world_simulation",
                        "launchFile": "empty_world.launch"
                    }
                }
            ],
            "tags": {}
        }
    ],
    "unprocessedJobs": []
}
```

- Pour plus de détails sur l'API, reportez-vous BatchDescribeSimulationJobà la section Référence des AWS CLI commandes.

`cancel-simulation-job`

L'exemple de code suivant montre comment utiliser`cancel-simulation-job`.

AWS CLI

Pour annuler une tâche de simulation

L'`cancel-simulation-job`exemple suivant annule la tâche de simulation spécifiée.

```
aws robomaker cancel-simulation-job \
```

```
    --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- Pour plus de détails sur l'API, reportez-vous CancelSimulationJobà la section Référence des AWS CLI commandes.

`create-deployment-job`

L'exemple de code suivant montre comment utiliser`create-deployment-job`.

AWS CLI

Pour créer une tâche de déploiement

Cet exemple crée une tâche de déploiement pour la flotte MyFleet. Il inclut une variable d'environnement nommée « ENVIRONNEMENT ». Il attache également une balise nommée « Région ».

Commande :

```
aws robomaker create-deployment-job --deployment-
config concurrentDeploymentPercentage=20,failureThresholdPercentage=25
 --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711 --tags Region=West --deployment-application-
configs application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
RoboMakerVoiceInteractionRobot/1546537110575,applicationVersion=1,launchConfig={environmentV
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/sim-0974h36s4v0t",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentApplicationConfigs": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerVoiceInteractionRobot/1546537110575",
          "applicationVersion": "1",
          "launchConfig": {
              "packageName": "voice_interaction_robot",
              "launchFile": "await_commands.launch",
              "environmentVariables": {
```

```
                    "ENVIRONMENT": "Beta"
                }
            }
        }
    ],
    "createdAt": 1550770236.0,
    "deploymentConfig": {
        "concurrentDeploymentPercentage": 20,
        "failureThresholdPercentage": 25
    },
    "tags": {
        "Region": "West"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDeploymentJobà la section Référence des AWS CLI commandes.

`create-fleet`

L'exemple de code suivant montre comment utiliser`create-fleet`.

AWS CLI

Pour créer une flotte

Cet exemple crée une flotte. Il attache un tag nommé Region.

Commande :

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

Sortie :

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyOtherFleet/1550771394395",
    "name": "MyFleet",
    "createdAt": 1550771394.0,
    "tags": {
        "Region": "East"
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFleetà la section Référence des AWS CLI commandes.

`create-robot-application-version`

L'exemple de code suivant montre comment utiliser`create-robot-application-version`.

AWS CLI

Pour créer une version d'une application de robot

Cet exemple crée une version d'une application de robot.

Commande :

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "1",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "etag": "f8cf5526f1c6e7b3a72c3ed3f79c5493-70",
          "architecture": "ARMHF"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "9986bb8d-a695-4ab4-8810-9f4a74d1aa00"
  "tags": {}
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRobotApplicationVersionà la section Référence des AWS CLI commandes.

`create-robot-application`

L'exemple de code suivant montre comment utiliser`create-robot-application`.

AWS CLI

Pour créer une application de robot

Cet exemple crée une application de robot.

Commande :

```
aws robomaker create-robot-application --name MyRobotApplication
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-robot-
application.tar.gz,architecture=X86_64 --robot-software-
suite name=ROS,version=Kinetic
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "ARMHF"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "1f3cb539-9239-4841-a656-d3efcffa07e1",
  "tags": {}
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRobotApplicationà la section Référence des AWS CLI commandes.

`create-robot`

L'exemple de code suivant montre comment utiliser`create-robot`.

AWS CLI

Pour créer un robot

Cet exemple crée un robot. Il utilise l'architecture ARMHF. Il attache également une balise nommée Region.

Commande :

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
  "createdAt": 1550772325.0,
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "architecture": "ARMHF",
  "tags": {
      "Region": "East"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRobotà la section Référence des AWS CLI commandes.

`create-simulation-application-version`

L'exemple de code suivant montre comment utiliser`create-simulation-application-version`.

AWS CLI

Pour créer une version d'application de simulation

Cet exemple crée une version d'une application de robot.

Commande :

```
aws robomaker create-simulation-application-version --
application arn:aws:robomaker:us-west-2:111111111111:robot-application/
MySimulationApplication/1551203427605
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MyRobotApplication/1551203427605",
  "name": "MyRobotApplication",
  "version": "1",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "etag": "00d8a94ff113856688c4fce618ae0f45-94",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
 "lastUpdatedAt": 1551203853.0,
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
  "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSimulationApplicationVersionà la section Référence des AWS CLI commandes.

`create-simulation-application`

L'exemple de code suivant montre comment utiliser`create-simulation-application`.

AWS CLI

Pour créer une application de simulation

Cet exemple crée une application de simulation.

Commande :

```
aws robomaker create-simulation-application  --name MyRobotApplication
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-simulation-
application.tar.gz,architecture=ARMHF --robot-software-
suite name=ROS,version=Kinetic --simulation-software-suite name=Gazebo,version=7 --
rendering-engine name=OGRE,version=1.x
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MyRobotApplication/1551203301792",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
```

```
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
  "lastUpdatedAt": 1551203301.0,
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
  "tags": {}
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSimulationApplicationà la section Référence des AWS CLI commandes.

`create-simulation-job`

L'exemple de code suivant montre comment utiliser`create-simulation-job`.

AWS CLI

Pour créer une tâche de simulation

Cet exemple crée une tâche de simulation. Il utilise une application robotisée et une application de simulation.

Commande :

```
aws robomaker create-simulation-job --max-job-duration-
in-seconds 3600 --iam-role arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G0OBWTQ8YBG6 --robot-
applications application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821,launchConfig={packageName=hello_world_robot,launchFile=rota
 --simulation-applications application=arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605,launchConfig={packageName=hello_world_simulation,launc
 --tags Region=North
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-w7m68wpr05h8",
  "status": "Pending",
```

```
  "lastUpdatedAt": 1551213837.0,
  "failureBehavior": "Fail",
  "clientRequestToken": "b283ccce-e468-43ee-8642-be76a9d69f15",
  "maxJobDurationInSeconds": 3600,
  "simulationTimeMillis": 0,
  "iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",
  "robotApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1551203485821",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_robot",
              "launchFile": "rotate.launch"
          }
      }
  ],
  "simulationApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-
application/MySimulationApplication/1551203427605",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_simulation",
              "launchFile": "empty_world.launch"
          }
      }
  ],
  "tags": {
      "Region": "North"
  }
}
```

- Pour plus de détails sur l'API, voir CreateSimulationJobla section Référence des AWS CLI commandes.

`delete-fleet`

L'exemple de code suivant montre comment utiliser`delete-fleet`.

AWS CLI

Pour supprimer une flotte

Cet exemple supprime une flotte.

Commande :

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- Pour plus de détails sur l'API, voir DeleteFleetla section Référence des AWS CLI commandes.

`delete-robot-application`

L'exemple de code suivant montre comment utiliser`delete-robot-application`.

AWS CLI

Pour supprimer une application de robot

Cet exemple supprime une application de robot.

Commande :

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- Pour plus de détails sur l'API, voir DeleteRobotApplicationla section Référence des AWS CLI commandes.

`delete-robot`

L'exemple de code suivant montre comment utiliser`delete-robot`.

AWS CLI

Pour supprimer un robot

Cet exemple supprime un robot.

Commande :

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540829698778
```

- Pour plus de détails sur l'API, voir DeleteRobotla section Référence des AWS CLI commandes.

`delete-simulation-application`

L'exemple de code suivant montre comment utiliser`delete-simulation-application`.

AWS CLI

Pour supprimer une application de simulation

Cet exemple supprime une application de simulation.

Commande :

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-
west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- Pour plus de détails sur l'API, voir DeleteSimulationApplicationla section Référence des AWS CLI commandes.

`deregister-robot`

L'exemple de code suivant montre comment utiliser`deregister-robot`.

AWS CLI

Pour désenregistrer un robot d'une flotte

Cet exemple permet de désenregistrer un robot d'une flotte.

Commande :

```
aws robomaker deregister-robot --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --
robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Sortie :

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
```

```
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- Pour plus de détails sur l'API, voir DeregisterRobotla section Référence des AWS CLI commandes.

`describe-deployment-job`

L'exemple de code suivant montre comment utiliser`describe-deployment-job`.

AWS CLI

Pour décrire une tâche de déploiement

L'`describe-deployment-job`exemple suivant récupère les détails de la tâche de déploiement spécifiée.

```
aws robomaker describe-deployment-job \
    --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn
```

Sortie :

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn",
    "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711",
    "status": "InProgress",
    "deploymentConfig": {
        "concurrentDeploymentPercentage": 20,
        "failureThresholdPercentage": 25
    },
    "deploymentApplicationConfigs": [
        {
            "application": "arn:aws:robomaker:us-west-2:111111111111:robot-application/RoboMakerHelloWorldRobot/1546541208251",
            "applicationVersion": "1",
            "launchConfig": {
                "packageName": "hello_world_robot",
                "launchFile": "rotate.launch"
            }
```

```
        }
    ],
    "createdAt": 1551218369.0,
    "robotDeploymentSummary": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540834232469",
            "deploymentStartTime": 1551218376.0,
            "status": "Deploying",
            "progressDetail": {}
        }
    ],
    "tags": {}
}
```

- Pour plus de détails sur l'API, voir DescribeDeploymentJobla section Référence des AWS CLI commandes.

`describe-fleet`

L'exemple de code suivant montre comment utiliser`describe-fleet`.

AWS CLI

Pour décrire une flotte

L'`describe-fleet`exemple suivant permet de récupérer les détails de la flotte spécifiée.

```
aws robomaker describe-fleet \
    --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907
```

Sortie :

```
{
    "name": "MyFleet",
    "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
    "robots": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540834232469",
```

```
            "createdAt": 1540834232.0
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyOtherRobot/1540829698778",
            "createdAt": 1540829698.0
        }
    ],
    "createdAt": 1539894765.0,
    "lastDeploymentStatus": "Succeeded",
    "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-xl8qssl6pbcn",
    "lastDeploymentTime": 1551218369.0,
    "tags": {}
}
```

- Pour plus de détails sur l'API, voir DescribeFleetla section Référence des AWS CLI commandes.

`describe-robot-application`

L'exemple de code suivant montre comment utiliser`describe-robot-application`.

AWS CLI

Pour décrire une application robotisée

Cet exemple décrit une application de robot.

Commande :

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
```

```
        {
            "s3Bucket": "amzn-s3-demo-bucket",
            "s3Key": "my-robot-application.tar.gz",
            "architecture": "X86_64"
        }
    ],
    "robotSoftwareSuite": {
        "name": "ROS",
        "version": "Kinetic"
    },
    "revisionId": "e72efe0d-f44f-4333-b604-f6fa5c6bb50b",
    "lastUpdatedAt": 1551203485.0,
    "tags": {}
}
```

- Pour plus de détails sur l'API, voir DescribeRobotApplicationla section Référence des AWS CLI commandes.

`describe-robot`

L'exemple de code suivant montre comment utiliser`describe-robot`.

AWS CLI

Pour décrire un robot

Cet exemple décrit un robot.

Commande :

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Sortie :

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
    "name": "MyRobot",
    "status": "Available",
    "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
    "createdAt": 1550772325.0,
    "architecture": "ARMHF",
```

```
  "tags": {
      "Region": "East"
  }
}
```

- Pour plus de détails sur l'API, voir DescribeRobotla section Référence des AWS CLI commandes.

`describe-simulation-application`

L'exemple de code suivant montre comment utiliser`describe-simulation-application`.

AWS CLI

Pour décrire une application de simulation

Cet exemple décrit une application de simulation.

Commande :

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
```

```
        "name": "ROS",
        "version": "Kinetic"
    },
    "renderingEngine": {
        "name": "OGRE",
        "version": "1.x"
    },
    "revisionId": "783674ab-b7b8-42d9-b01f-9373907987e5",
    "lastUpdatedAt": 1551203427.0,
    "tags": {}
}
```

- Pour plus de détails sur l'API, voir DescribeSimulationApplicationla section Référence des AWS CLI commandes.

`describe-simulation-job`

L'exemple de code suivant montre comment utiliser`describe-simulation-job`.

AWS CLI

Pour décrire une tâche de simulation

Cet exemple décrit une tâche de simulation.

Commande :

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

Sortie :

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6",
    "status": "Running",
    "lastUpdatedAt": 1551219349.0,
    "failureBehavior": "Continue",
    "clientRequestToken": "a19ec4b5-e50d-3591-33da-c2e593c60615",
    "outputLocation": {
        "s3Bucket": "my-output-bucket",
        "s3Prefix": "output"
    },
```

```
  "maxJobDurationInSeconds": 3600,
  "simulationTimeMillis": 0,
  "iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",
  "robotApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1551206341136",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_robot",
              "launchFile": "rotate.launch"
          }
      }
  ],
  "simulationApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-
application/MySimulationApplication/1551206347967",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_simulation",
              "launchFile": "empty_world.launch"
          }
      }
  ],
  "tags": {}
}
```

- Pour plus de détails sur l'API, voir DescribeSimulationJobla section Référence des AWS CLI commandes.

`list-deployment-jobs`

L'exemple de code suivant montre comment utiliser`list-deployment-jobs`.

AWS CLI

Pour répertorier les tâches de déploiement

L'`list-deployment-jobs`exemple suivant extrait une liste de tâches de déploiement.

```
aws robomaker list-deployment-jobs
```

Sortie :

```
{
    "deploymentJobs": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
sim-6293szzm56rv",
            "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
            "status": "InProgress",
            "deploymentApplicationConfigs": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/HelloWorldRobot/1546537110575",
                    "applicationVersion": "1",
                    "launchConfig": {
                        "packageName": "hello_world_robot",
                        "launchFile": "rotate.launch",
                        "environmentVariables": {
                            "ENVIRONMENT": "Desert"
                        }
                    }
                }
            ],
            "deploymentConfig": {
                "concurrentDeploymentPercentage": 20,
                "failureThresholdPercentage": 25
            },
            "createdAt": 1550689373.0
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-4w4g69p25zdb",
            "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
            "status": "Pending",
            "deploymentApplicationConfigs": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerHelloWorld-1544562726923_YGHM_sh5M/1544562822877",
                    "applicationVersion": "1",
                    "launchConfig": {
                        "packageName": "fail",
                        "launchFile": "fail"
```

```
                    }
                }
            ],
            "deploymentConfig": {
                "concurrentDeploymentPercentage": 20,
                "failureThresholdPercentage": 25
            },
            "failureReason": "",
            "failureCode": "",
            "createdAt": 1544719763.0
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListDeploymentJobsla section Référence des AWS CLI commandes.

`list-fleets`

L'exemple de code suivant montre comment utiliser`list-fleets`.

AWS CLI

Pour répertorier les flottes

Cet exemple répertorie les flottes. Un maximum de 20 flottes seront retournées.

Commande :

```
aws robomaker list-fleets --max-items 20
```

Sortie :

```
{
  "fleetDetails": [
      {
          "name": "Trek",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
          "createdAt": 1539894765.0,
          "lastDeploymentStatus": "Failed",
          "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-
job/deployment-4w4g69p25zdb",
```

```
            "lastDeploymentTime": 1544719763.0
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListFleetsla section Référence des AWS CLI commandes.

`list-robot-applications`

L'exemple de code suivant montre comment utiliser`list-robot-applications`.

AWS CLI

Pour répertorier les applications de robots

Cet exemple répertorie les applications de robots. Les résultats sont limités à 20 applications robotisées.

Commande :

```
aws robomaker list-robot-applications --max-results 20
```

Sortie :

```
{
  "robotApplicationSummaries": [
      {
          "name": "MyRobot",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobot/1546537110575",
          "version": "$LATEST",
          "lastUpdatedAt": 1546540372.0
      },
      {
          "name": "AnotherRobot",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
AnotherRobot/1546541208251",
          "version": "$LATEST",
          "lastUpdatedAt": 1546541208.0
      },
      {
          "name": "MySuperRobot",
```

```
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MySuperRobot/1547663517377",
            "version": "$LATEST",
            "lastUpdatedAt": 1547663517.0
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListRobotApplicationsla section Référence des AWS CLI commandes.

`list-robots`

L'exemple de code suivant montre comment utiliser`list-robots`.

AWS CLI

Pour répertorier les robots

Cet exemple répertorie les robots. Un maximum de 20 robots seront renvoyés.

Commande :

```
aws robomaker list-robots --max-results 20
```

Sortie :

```
{
    "robots": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot100/1544035373264",
            "name": "Robot100",
            "status": "Available",
            "createdAt": 1544035373.0,
            "architecture": "X86_64"
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot101/1542146976587",
            "name": "Robot101",
            "status": "Available",
            "createdAt": 1542146976.0,
```

```
            "architecture": "X86_64"
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot102/1540834232469",
            "name": "Robot102",
            "fleetArn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711",
            "status": "Available",
            "createdAt": 1540834232.0,
            "architecture": "X86_64",
            "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-
job/deployment-jb007b75gl5f",
            "lastDeploymentTime": 1550689533.0
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540829698778",
            "name": "MyRobot",
            "status": "Registered",
            "createdAt": 1540829698.0,
            "architecture": "X86_64"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListRobotsla section Référence des AWS CLI commandes.

`list-simulation-applications`

L'exemple de code suivant montre comment utiliser`list-simulation-applications`.

AWS CLI

Pour répertorier les applications de simulation

Cet exemple répertorie les applications de simulation. Un maximum de 20 applications de simulation seront renvoyées.

Commande :

```
aws robomaker list-simulation-applications --max-results 20
```

Sortie :


```
{
  "simulationApplicationSummaries": [
      {
          "name": "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
          "version": "$LATEST",
          "lastUpdatedAt": 1548959170.0
      },
      {
          "name": "RoboMakerHelloWorldSimulation",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
          "version": "$LATEST",
          "lastUpdatedAt": 1546541198.0
      },
      {
          "name": "RoboMakerObjectTrackerSimulation",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerObjectTrackerSimulation/1545846795615",
          "version": "$LATEST",
          "lastUpdatedAt": 1545847405.0
      },
      {
          "name": "RoboMakerVoiceInteractionSimulation",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerVoiceInteractionSimulation/1546537100507",
          "version": "$LATEST",
          "lastUpdatedAt": 1546540352.0
      },
      {
          "name": "AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
          "version": "$LATEST",
          "lastUpdatedAt": 1547663521.0
      },
      {
          "name": "AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-/1545848370525",
          "version": "$LATEST",
```

```
            "lastUpdatedAt": 1545848370.0
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListSimulationApplicationsla section Référence des AWS CLI commandes.

`list-simulation-jobs`

L'exemple de code suivant montre comment utiliser`list-simulation-jobs`.

AWS CLI

Pour répertorier les tâches de simulation

Cet exemple répertorie les tâches de simulation.

Commande :

```
aws robomaker list-simulation-jobs
```

Sortie :

```
{
  "simulationJobSummaries": [
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x",
          "lastUpdatedAt": 1548959178.0,
          "status": "Completed",
          "simulationApplicationNames": [
              "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq"
          ],
          "robotApplicationNames": [
              null
          ]
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
b27c4rkrtzcw",
          "lastUpdatedAt": 1543514088.0,
          "status": "Canceled",
```

```
            "simulationApplicationNames": [
                "AWSRoboMakerPersonDetection-1543513948280_T8rHW2_lu"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerPersonDetection-1543513948280_EYaMT0mYb"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-51vxjbzy4q8t",
            "lastUpdatedAt": 1543508858.0,
            "status": "Canceled",
            "simulationApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_lFF9ZQyx6"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
kgf1fqxflqbx",
            "lastUpdatedAt": 1543504862.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_lFF9ZQyx6"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
vw8lvh061nqt",
            "lastUpdatedAt": 1543441430.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341__yb_Jg96l"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
            ]
        },
        {
```

```
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
txy5ypxmhz84",
            "lastUpdatedAt": 1543437488.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341__yb_Jg96l"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListSimulationJobsla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

Cet exemple répertorie les balises d'une AWS RoboMaker ressource.

Commande :

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```

Sortie :

```
{
    "tags": {
        "Region": "North",
        "Stage": "Initial"
    }
}
```

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`register-robot`

L'exemple de code suivant montre comment utiliser`register-robot`.

AWS CLI

Pour enregistrer un robot

Cet exemple enregistre un robot dans une flotte.

Commande :

```
aws robomaker register-robot --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --
robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Sortie :

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- Pour plus de détails sur l'API, voir RegisterRobotla section Référence des AWS CLI commandes.

`restart-simulation-job`

L'exemple de code suivant montre comment utiliser`restart-simulation-job`.

AWS CLI

Pour redémarrer une simulation

Cet exemple redémarre une simulation.

Commande :

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- Pour plus de détails sur l'API, voir RestartSimulationJobla section Référence des AWS CLI commandes.

`sync-deployment-job`

L'exemple de code suivant montre comment utiliser`sync-deployment-job`.

AWS CLI

Pour synchroniser une tâche de déploiement

Cet exemple synchronise une tâche de déploiement.

Commande :

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/Trek/1539894765711
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-09ccxs3tlfms",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentConfig": {
      "concurrentDeploymentPercentage": 20,
      "failureThresholdPercentage": 25
  },
  "deploymentApplicationConfigs": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1546541208251",
          "applicationVersion": "1",
          "launchConfig": {
              "packageName": "hello_world_simulation",
              "launchFile": "empty_world.launch"
```

```
            }
        }
    ],
    "createdAt": 1551286954.0
}
```

- Pour plus de détails sur l'API, voir SyncDeploymentJobla section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utilisertag-resource.

AWS CLI

Pour étiqueter une ressource

Cet exemple balise une ressource. Il attache deux balises : Region et Stage.

Commande :

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliseruntag-resource.

AWS CLI

Pour annuler le balisage d'une ressource

Cet exemple supprime une balise d'une ressource. Il supprime le tag Region.

Commande :

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-robot-application`

L'exemple de code suivant montre comment utiliser`update-robot-application`.

AWS CLI

Pour mettre à jour une application de robot

Cet exemple met à jour une application de robot.

Commande :

```
aws robomaker update-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-robot-
application.tar.gz,architecture=X86_64 --robot-software-
suite name=ROS,version=Kinetic
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551287993.0,
  "revisionId": "20b5e331-24fd-4504-8b8c-531afe5f4c94"
}
```

- Pour plus de détails sur l'API, voir UpdateRobotApplicationla section Référence des AWS CLI commandes.

`update-simulation-application`

L'exemple de code suivant montre comment utiliserupdate-simulation-application.

AWS CLI

Pour mettre à jour une application de simulation

Cet exemple met à jour une application de simulation.

Commande :

```
aws robomaker update-simulation-application --application arn:aws:robomaker:us-
west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-simulation-
application.tar.gz,architecture=X86_64 --robot-software-
suite name=ROS,version=Kinetic --simulation-software-suite name=Gazebo,version=7 --
rendering-engine name=OGRE,version=1.x
```

Sortie :

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
```

```
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
  "lastUpdatedAt": 1551289361.0,
  "revisionId": "4a22cb5d-93c5-4cef-9311-52bdd119b79e"
}
```

- Pour plus de détails sur l'API, voir UpdateSimulationApplicationla section Référence des AWS CLI commandes.

Exemples de Route 53 utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Route 53.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`change-resource-record-sets`

L'exemple de code suivant montre comment utiliser`change-resource-record-sets`.

AWS CLI

Pour créer, mettre à jour ou supprimer un jeu d'enregistrements de ressources

La `change-resource-record-sets` commande suivante crée un ensemble d'enregistrements de ressources à l'aide de `hosted-zone-id` `Z1R8UBAEXAMPLE` la configuration au format JSON figurant dans le fichier : `C:\awscli\route53\change-resource-record-sets.json`

```
aws route53 change-resource-record-sets --hosted-zone-id Z1R8UBAEXAMPLE --change-
batch file://C:\awscli\route53\change-resource-record-sets.json
```

Pour plus d'informations, consultez POST ChangeResourceRecordSets dans le manuel Amazon Route 53 API Reference.

La configuration du fichier JSON dépend du type d'ensemble d'enregistrements de ressources que vous souhaitez créer :

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover Alias

Syntaxe de base :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ]
      }
    },
    {...}
  ]
}
```

Syntaxe pondérée :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
```

```
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

Syntaxe des alias :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

Syntaxe d'alias pondérée :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

Syntaxe de latence :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
```

```
          },
          {...}
        ],
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

Syntaxe de l'alias de latence :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

Syntaxe de basculement :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
```

```
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Failover": "PRIMARY" | "SECONDARY",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
        "HealthCheckId": "ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

Syntaxe de l'alias Failover :

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Failover": "PRIMARY" | "SECONDARY",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
```

```
    {...}
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous ChangeResourceRecordSetsà la section Référence des AWS CLI commandes.

`change-tags-for-resource`

L'exemple de code suivant montre comment utiliser`change-tags-for-resource`.

AWS CLI

La commande suivante ajoute une balise nommée `owner` à une ressource de contrôle de santé spécifiée par son ID :

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-
id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner,Value=myboss
```

La commande suivante supprime une balise nommée `owner` à partir d'une ressource de zone hébergée spécifiée par un ID :

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-
id Z1523434445 --remove-tag-keys owner
```

- Pour plus de détails sur l'API, reportez-vous ChangeTagsForResourceà la section Référence des AWS CLI commandes.

`create-health-check`

L'exemple de code suivant montre comment utiliser`create-health-check`.

AWS CLI

Pour créer un bilan de santé

La `create-health-check` commande suivante crée un contrôle de santé à l'aide de la référence de l'appelant `2014-04-01-18:47` et de la configuration au format JSON du fichier : `C:\awscli\route53\create-health-check.json`

```
aws route53 create-health-check --caller-reference 2014-04-01-18:47 --health-check-
config file://C:\awscli\route53\create-health-check.json
```

Syntaxe JSON :

```
{
  "IPAddress": "IP address of the endpoint to check",
  "Port": port on the endpoint to check--required when Type is "TCP",
  "Type": "HTTP"|"HTTPS"|"HTTP_STR_MATCH"|"HTTPS_STR_MATCH"|"TCP",
  "ResourcePath": "path of the file that you want Amazon Route 53 to request--all
 Types except TCP",
  "FullyQualifiedDomainName": "domain name of the endpoint to check--all Types
 except TCP",
  "SearchString": "if Type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string to
 search for in the response body from the specified resource",
  "RequestInterval": 10 | 30,
  "FailureThreshold": integer between 1 and 10
}
```

Pour ajouter le bilan de santé à un ensemble d'enregistrements de ressources Route 53, utilisez la `change-resource-record-sets` commande.

Pour plus d'informations, consultez Amazon Route 53 Health Checks and DNS Failover dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateHealthCheckà la section Référence des AWS CLI commandes.

`create-hosted-zone`

L'exemple de code suivant montre comment utiliser`create-hosted-zone`.

AWS CLI

Pour créer une zone hébergée

La `create-hosted-zone` commande suivante ajoute une zone hébergée nommée à `example.com` l'aide de la référence `2014-04-01-18:47` de l'appelant. Le commentaire facultatif inclut un espace, il doit donc être placé entre guillemets :

```
aws route53 create-hosted-zone --name example.com --caller-
reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

Pour plus d'informations, consultez la section Travailler avec des zones hébergées dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous CreateHostedZoneà la section Référence des AWS CLI commandes.

`delete-health-check`

L'exemple de code suivant montre comment utiliser`delete-health-check`.

AWS CLI

Pour supprimer un bilan de santé

La `delete-health-check` commande suivante supprime le bilan de santé avec un `health-check-id` de `e75b48d9-547a-4c3d-88a5-ae4002397608` :

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-
ae4002397608
```

- Pour plus de détails sur l'API, reportez-vous DeleteHealthCheckà la section Référence des AWS CLI commandes.

`delete-hosted-zone`

L'exemple de code suivant montre comment utiliser`delete-hosted-zone`.

AWS CLI

Pour supprimer une zone hébergée

La `delete-hosted-zone` commande suivante supprime la zone hébergée avec un `id` de `Z36KTIQEXAMPLE` :

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous DeleteHostedZoneà la section Référence des AWS CLI commandes.

`get-change`

L'exemple de code suivant montre comment utiliser`get-change`.

AWS CLI

Pour obtenir le statut d'une modification apportée aux ensembles d'enregistrements de ressources

La `get-change` commande suivante permet d'obtenir le statut et d'autres informations relatives à la `change-resource-record-sets` demande dont le numéro est le suivant `/change/CWPIK4URU2I5S` : `Id`

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- Pour plus de détails sur l'API, reportez-vous GetChangeà la section Référence des AWS CLI commandes.

`get-health-check`

L'exemple de code suivant montre comment utiliser`get-health-check`.

AWS CLI

Pour obtenir des informations sur un bilan de santé

La `get-health-check` commande suivante permet d'obtenir des informations sur le bilan de santé qui comporte un `health-check-id` de `02ec8401-9879-4259-91fa-04e66d094674` :

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- Pour plus de détails sur l'API, reportez-vous GetHealthCheckà la section Référence des AWS CLI commandes.

`get-hosted-zone`

L'exemple de code suivant montre comment utiliser`get-hosted-zone`.

AWS CLI

Pour obtenir des informations sur une zone hébergée

La `get-hosted-zone` commande suivante permet d'obtenir des informations sur la zone hébergée avec un `id` de `Z1R8UBAEXAMPLE` :

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous GetHostedZoneà la section Référence des AWS CLI commandes.

`list-health-checks`

L'exemple de code suivant montre comment utiliser`list-health-checks`.

AWS CLI

Pour répertorier les bilans de santé associés au AWS compte courant

La `list-health-checks` commande suivante répertorie des informations détaillées sur les 100 premiers bilans de santé associés au AWS compte courant. :

```
aws route53 list-health-checks
```

Si vous avez plus de 100 bilans de santé, ou si vous souhaitez les répertorier par groupes inférieurs à 100, incluez le `--maxitems` paramètre. Par exemple, pour répertorier les bilans de santé un par un, utilisez la commande suivante :

```
aws route53 list-health-checks --max-items 1
```

Pour afficher le bilan de santé suivant, prenez la valeur `NextToken` de la réponse à la commande précédente et incluez-la dans le `--starting-token` paramètre, par exemple :

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous ListHealthChecksà la section Référence des AWS CLI commandes.

`list-hosted-zones-by-name`

L'exemple de code suivant montre comment utiliser`list-hosted-zones-by-name`.

AWS CLI

La commande suivante répertorie jusqu'à 100 zones hébergées classées par nom de domaine :

```
aws route53 list-hosted-zones-by-name
```

Sortie :

```
{
  "HostedZones": [
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "test20150527-2",
          "Config": {
              "Comment": "test2",
              "PrivateZone": false
          },
          "Id": "/hostedzone/Z119WBBTVP5WFX",
          "Name": "2.example.com."
      },
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "test20150527-1",
          "Config": {
              "Comment": "test",
              "PrivateZone": false
          },
          "Id": "/hostedzone/Z3P5QSUBK4POTI",
          "Name": "www.example.com."
      }
  ],
  "IsTruncated": false,
  "MaxItems": "100"
}
```

La commande suivante répertorie les zones hébergées classées par nom, en commençant par `www.example.com` :

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

Sortie :

```
{
  "HostedZones": [
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "mwunderl20150527-1",
          "Config": {
              "Comment": "test",
              "PrivateZone": false
          },
          "Id": "/hostedzone/Z3P5QSUBK4POTI",
          "Name": "www.example.com."
      }
  ],
  "DNSName": "www.example.com",
  "IsTruncated": false,
  "MaxItems": "100"
}
```

- Pour plus de détails sur l'API, reportez-vous ListHostedZonesByNameà la section Référence des AWS CLI commandes.

`list-hosted-zones`

L'exemple de code suivant montre comment utiliser`list-hosted-zones`.

AWS CLI

Pour répertorier les zones hébergées associées au AWS compte actuel

La `list-hosted-zones` commande suivante répertorie les informations récapitulatives sur les 100 premières zones hébergées associées au AWS compte actuel. :

```
aws route53 list-hosted-zones
```

Si vous avez plus de 100 zones hébergées ou si vous souhaitez les répertorier dans des groupes inférieurs à 100, incluez le paramètre `--max-items`. Par exemple, pour répertorier les zones hébergées une par une, utilisez la commande suivante :

```
aws route53 list-hosted-zones --max-items 1
```

Pour afficher les informations relatives à la zone hébergée suivante, prenez la valeur `NextToken` de la réponse à la commande précédente et incluez-la dans le paramètre `--starting-token`, par exemple :

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- Pour plus de détails sur l'API, reportez-vous ListHostedZonesà la section Référence des AWS CLI commandes.

`list-query-logging-configs`

L'exemple de code suivant montre comment utiliser`list-query-logging-configs`.

AWS CLI

Pour répertorier les configurations de journalisation des requêtes

L'`list-query-logging-configs`exemple suivant répertorie les informations relatives aux 100 premières configurations de journalisation des requêtes de votre AWS compte, pour la zone hébergéeZ1OX3WQEXAMPLE.

```
aws route53 list-query-logging-configs \
    --hosted-zone-id Z1OX3WQEXAMPLE
```

Sortie :

```
{
    "QueryLoggingConfigs": [
        {
            "Id": "964ff34e-ae03-4f06-80a2-9683cexample",
            "HostedZoneId": "Z1OX3WQEXAMPLE",
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/route53/example.com:*"
        }
    ]
}
```

Pour plus d'informations, consultez la section Journalisation des requêtes DNS dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListQueryLoggingConfigsà la section Référence des AWS CLI commandes.

`list-resource-record-sets`

L'exemple de code suivant montre comment utiliser`list-resource-record-sets`.

AWS CLI

Pour répertorier les ensembles d'enregistrements de ressources dans une zone hébergée

La `list-resource-record-sets` commande suivante répertorie des informations récapitulatives sur les 100 premiers ensembles d'enregistrements de ressources dans une zone hébergée spécifiée. :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

Si la zone hébergée contient plus de 100 ensembles d'enregistrements de ressources, ou si vous souhaitez les répertorier par groupes inférieurs à 100, incluez le `--maxitems` paramètre. Par exemple, pour répertorier les ensembles d'enregistrements de ressources un par un, utilisez la commande suivante :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

Pour afficher les informations relatives au prochain enregistrement de ressource défini dans la zone hébergée, prenez la valeur `NextToken` de la réponse à la commande précédente et incluez-la dans le `--starting-token` paramètre, par exemple :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
 --starting-token Z3M3LMPEXAMPLE
```

Pour afficher tous les ensembles d'enregistrements de ressources portant un nom particulier, utilisez le `--query` paramètre pour les filtrer. Par exemple :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --
query "ResourceRecordSets[?Name == 'example.domain.']"
```

- Pour plus de détails sur l'API, reportez-vous ListResourceRecordSetsà la section Référence des AWS CLI commandes.

Exemples d'enregistrement de domaine Route 53 à l'aide de AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de l'enregistrement de domaine AWS Command Line Interface avec Route 53.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`check-domain-availability`

L'exemple de code suivant montre comment utiliser`check-domain-availability`.

AWS CLI

Pour déterminer si vous pouvez enregistrer un nom de domaine avec Route 53

La `check-domain-availability` commande suivante renvoie des informations indiquant si le nom de domaine `example.com` est disponible pour être enregistré via Route 53.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains check-domain-availability \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "Availability": "UNAVAILABLE"
```

```
}
```

Route 53 prend en charge un grand nombre de domaines de premier niveau (TLDs), tels que `.com` et `.jp`, mais nous ne prenons pas en charge tous les domaines disponibles TLDs. Si vous vérifiez la disponibilité d'un domaine et que Route 53 ne prend pas en charge le TLD, `check-domain-availability` renvoie le message suivant.

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability
 operation: <top-level domain> tld is not supported.
```

Pour obtenir la liste des domaines TLDs que vous pouvez utiliser lors de l'enregistrement d'un domaine auprès de Route 53, consultez la section Domaines que vous pouvez enregistrer auprès d'Amazon Route 53 dans le guide du développeur Amazon Route 53. Pour plus d'informations sur l'enregistrement de domaines auprès d'Amazon Route 53, consultez la section Enregistrer un nouveau domaine dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous CheckDomainAvailabilityà la section Référence des AWS CLI commandes.

`check-domain-transferability`

L'exemple de code suivant montre comment utiliser`check-domain-transferability`.

AWS CLI

Pour déterminer si un domaine peut être transféré vers Route 53

La `check-domain-transferability` commande suivante renvoie des informations indiquant si vous pouvez transférer le nom de domaine `example.com` vers Route 53.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains check-domain-transferability \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
```

```
    "Transferability": {
        "Transferable": "UNTRANSFERABLE"
    }
}
```

Pour plus d'informations, consultez la section Transfert de l'enregistrement d'un domaine vers Amazon Route 53 dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous CheckDomainTransferabilityà la section Référence des AWS CLI commandes.

`delete-tags-for-domain`

L'exemple de code suivant montre comment utiliser`delete-tags-for-domain`.

AWS CLI

Pour supprimer les balises d'un domaine

La `delete-tags-for-domain` commande suivante supprime trois balises du domaine spécifié. Notez que vous ne spécifiez que la clé de balise, pas la valeur de la balise.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains delete-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com \
    --tags-to-delete accounting-key hr-key engineering-key
```

Cette commande ne produit aucun résultat.

Pour confirmer que les balises ont été supprimées, vous pouvez exécuter list-tags-for-domain. Pour plus d'informations, consultez la section Marquage des ressources Amazon Route 53 dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous DeleteTagsForDomainà la section Référence des AWS CLI commandes.

`disable-domain-auto-renew`

L'exemple de code suivant montre comment utiliser`disable-domain-auto-renew`.

AWS CLI

Pour désactiver le renouvellement automatique d'un domaine

La `disable-domain-auto-renew` commande suivante configure Route 53 pour ne pas renouveler automatiquement le domaine `example.com` avant l'expiration de son enregistrement.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains disable-domain-auto-renew \
    --region us-east-1 \
    --domain-name example.com
```

Cette commande ne produit aucun résultat.

Pour confirmer que le paramètre a été modifié, vous pouvez exécuter get-domain-detail. Si le renouvellement automatique est désactivé, la valeur de `AutoRenew` est`False`. Pour plus d'informations sur le renouvellement automatique, consultez Renouveler l'enregistrement pour un domaine < https://docs.aws.amazon.com/Route53/ latest/DeveloperGuide/domain -renew.html dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DisableDomainAutoRenewà la section Référence des AWS CLI commandes.

`disable-domain-transfer-lock`

L'exemple de code suivant montre comment utiliser`disable-domain-transfer-lock`.

AWS CLI

Pour désactiver le verrouillage des transferts sur un domaine

La `disable-domain-transfer-lock` commande suivante supprime le verrou de transfert sur le domaine `example.com` afin que celui-ci puisse être transféré vers un autre bureau d'enregistrement. Cette commande modifie le `clientTransferProhibited` statut.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains disable-domain-transfer-lock \
```

```
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

Pour confirmer que le verrou de transfert a été modifié, vous pouvez exécuter get-domain-detail. Lorsque le verrou de transfert est désactivé, la valeur de `StatusList` n'inclut pas`clientTransferProhibited`.

Pour plus d'informations sur le processus de transfert, consultez la section Transfert d'un domaine d'Amazon Route 53 vers un autre bureau d'enregistrement dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous DisableDomainTransferLockà la section Référence des AWS CLI commandes.

`enable-domain-auto-renew`

L'exemple de code suivant montre comment utiliser`enable-domain-auto-renew`.

AWS CLI

Pour activer le renouvellement automatique d'un domaine

La `enable-domain-auto-renew` commande suivante configure Route 53 pour qu'elle renouvelle automatiquement le domaine `example.com` avant l'expiration de son enregistrement.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains enable-domain-auto-renew \
    --region us-east-1 \
    --domain-name example.com
```

Cette commande ne produit aucun résultat. Pour confirmer que le paramètre a été modifié, vous pouvez exécuter get-domain-detail. Si le renouvellement automatique est activé, la valeur de `AutoRenew` est`True`.

Pour plus d'informations sur le renouvellement automatique, consultez Renouveler l'enregistrement pour un domaine < https://docs.aws.amazon.com/Route53/ latest/ DeveloperGuide/domain -renew.html dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous EnableDomainAutoRenewà la section Référence des AWS CLI commandes.

`enable-domain-transfer-lock`

L'exemple de code suivant montre comment utiliser`enable-domain-transfer-lock`.

AWS CLI

Pour activer le verrouillage des transferts sur un domaine

La `enable-domain-transfer-lock` commande suivante verrouille le domaine spécifié afin qu'il ne puisse pas être transféré vers un autre bureau d'enregistrement. Cette commande modifie le `clientTransferProhibited` statut.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains enable-domain-transfer-lock \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

Pour confirmer que le verrou de transfert a été modifié, vous pouvez exécuter get-domain-detail. Lorsque le verrou de transfert est activé, la valeur de `StatusList` inclut`clientTransferProhibited`.

Pour plus d'informations sur le processus de transfert, consultez la section Transfert d'un domaine d'Amazon Route 53 vers un autre bureau d'enregistrement dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous EnableDomainTransferLockà la section Référence des AWS CLI commandes.

`get-contact-reachability-status`

L'exemple de code suivant montre comment utiliser`get-contact-reachability-status`.

AWS CLI

Pour déterminer si le contact du titulaire a répondu à un e-mail de confirmation

La `get-contact-reachability-status` commande suivante renvoie des informations indiquant si le contact du titulaire pour le domaine spécifié a répondu à un e-mail de confirmation.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains get-contact-reachability-status \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "domainName": "example.com",
    "status": "DONE"
}
```

Pour plus d'informations, consultez la section Renvoyer des e-mails d'autorisation et de confirmation dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous GetContactReachabilityStatusà la section Référence des AWS CLI commandes.

`get-domain-detail`

L'exemple de code suivant montre comment utiliser`get-domain-detail`.

AWS CLI

Pour obtenir des informations détaillées sur un domaine spécifique

La `get-domain-detail` commande suivante affiche des informations détaillées sur le domaine spécifié.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains get-domain-detail \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "DomainName": "example.com",
    "Nameservers": [
        {
            "Name": "ns-2048.awsdns-64.com",
            "GlueIps": []
        },
        {
            "Name": "ns-2049.awsdns-65.net",
            "GlueIps": []
        },
        {
            "Name": "ns-2050.awsdns-66.org",
            "GlueIps": []
        },
        {
            "Name": "ns-2051.awsdns-67.co.uk",
            "GlueIps": []
        }
    ],
    "AutoRenew": true,
    "AdminContact": {
        "FirstName": "Saanvi",
        "LastName": "Sarkar",
        "ContactType": "COMPANY",
        "OrganizationName": "Example",
        "AddressLine1": "123 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "ssarkar@example.com",
        "ExtraParams": []
```

```
    },
    "RegistrantContact": {
        "FirstName": "Alejandro",
        "LastName": "Rosalez",
        "ContactType": "COMPANY",
        "OrganizationName": "Example",
        "AddressLine1": "123 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "arosalez@example.com",
        "ExtraParams": []
    },
    "TechContact": {
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "ContactType": "COMPANY",
        "OrganizationName": "Example",
        "AddressLine1": "123 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "wxiulan@example.com",
        "ExtraParams": []
    },
    "AdminPrivacy": true,
    "RegistrantPrivacy": true,
    "TechPrivacy": true,
    "RegistrarName": "Amazon Registrar, Inc.",
    "WhoIsServer": "whois.registrar.amazon",
    "RegistrarUrl": "http://registrar.amazon.com",
    "AbuseContactEmail": "abuse@registrar.amazon.com",
    "AbuseContactPhone": "+1.2062661000",
    "CreationDate": 1444934889.601,
    "ExpirationDate": 1602787689.0,
    "StatusList": [
        "clientTransferProhibited"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDomainDetailà la section Référence des AWS CLI commandes.

`get-domain-suggestions`

L'exemple de code suivant montre comment utiliser`get-domain-suggestions`.

AWS CLI

Pour obtenir une liste des noms de domaine suggérés

La `get-domain-suggestions` commande suivante affiche une liste de noms de domaine suggérés en fonction du nom de domaine`example.com`. La réponse inclut uniquement les noms de domaine disponibles. Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains get-domain-suggestions \
    --region us-east-1 \
    --domain-name example.com \
    --suggestion-count 10 \
    --only-available
```

Sortie :

```
{
    "SuggestionsList": [
        {
            "DomainName": "egzaampal.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplelaw.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplehouse.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "homeexample.net",
            "Availability": "AVAILABLE"
```

```
        },
        {
            "DomainName": "examplelist.com",
            "Availability": "AVAILABLE"
       },
        {
            "DomainName": "examplenews.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "officeexample.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "exampleworld.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "exampleart.com",
            "Availability": "AVAILABLE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetDomainSuggestionsà la section Référence des AWS CLI commandes.

`get-operation-detail`

L'exemple de code suivant montre comment utiliser`get-operation-detail`.

AWS CLI

Pour obtenir le statut actuel d'une opération

Certaines opérations d'enregistrement de domaine fonctionnent de manière asynchrone et renvoient une réponse avant leur fin. Ces opérations renvoient un identifiant d'opération que vous pouvez utiliser pour obtenir le statut actuel. La `get-operation-detail` commande suivante renvoie le statut de l'opération spécifiée.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains get-operation-detail \
    --region us-east-1 \
    --operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

Sortie :

```
{
    "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",
    "Status": "SUCCESSFUL",
    "DomainName": "example.com",
    "Type": "DOMAIN_LOCK",
    "SubmittedDate": 1573749367.864
}
```

- Pour plus de détails sur l'API, reportez-vous GetOperationDetailà la section Référence des AWS CLI commandes.

`list-domains`

L'exemple de code suivant montre comment utiliser`list-domains`.

AWS CLI

Pour répertorier les domaines enregistrés avec le AWS compte actuel

La `list-domains` commande suivante répertorie les informations récapitulatives sur les domaines enregistrés avec le AWS compte actuel.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains list-domains
    --region us-east-1
```

Sortie :

```
{
    "Domains": [
        {
            "DomainName": "example.com",
            "AutoRenew": true,
```

```
            "TransferLock": true,
            "Expiry": 1602712345.0
        },
        {
            "DomainName": "example.net",
            "AutoRenew": true,
            "TransferLock": true,
            "Expiry": 1602723456.0
        },
        {
            "DomainName": "example.org",
            "AutoRenew": true,
            "TransferLock": true,
            "Expiry": 1602734567.0
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListDomainsà la section Référence des AWS CLI commandes.

`list-operations`

L'exemple de code suivant montre comment utiliser`list-operations`.

AWS CLI

Pour répertorier le statut des opérations qui renvoient un ID d'opération

Certaines opérations d'enregistrement de domaine s'exécutent de manière asynchrone et renvoient une réponse avant leur fin. Ces opérations renvoient un identifiant d'opération que vous pouvez utiliser pour obtenir le statut actuel. La `list-operations` commande suivante répertorie les informations récapitulatives, y compris le statut, concernant les opérations d'enregistrement de domaine en cours.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains list-operations
    --region us-east-1
```

Sortie :

```
{
    "Operations": [
        {
            "OperationId": "aab9822f-1da0-4bf3-8a15-fd4e0example",
            "Status": "SUCCESSFUL",
            "Type": "DOMAIN_LOCK",
            "SubmittedDate": 1455321739.986
        },
        {
            "OperationId": "c24379ed-76be-42f8-bdad-9379bexample",
            "Status": "SUCCESSFUL",
            "Type": "UPDATE_NAMESERVER",
            "SubmittedDate": 1468960475.109
        },
        {
            "OperationId": "f47e1297-ef9e-4c2b-ae1e-a5fcbexample",
            "Status": "SUCCESSFUL",
            "Type": "RENEW_DOMAIN",
            "SubmittedDate": 1473561835.943
        },
        {
            "OperationId": "75584f23-b15f-459e-aed7-dc6f5example",
            "Status": "SUCCESSFUL",
            "Type": "UPDATE_DOMAIN_CONTACT",
            "SubmittedDate": 1547501003.41
        }
    ]
}
```

La sortie inclut toutes les opérations qui renvoient un ID d'opération et que vous avez effectuées sur tous les domaines que vous avez enregistrés à l'aide du AWS compte courant. Si vous souhaitez obtenir uniquement les opérations que vous avez soumises après une date spécifiée, vous pouvez inclure le `submitted-since` paramètre et spécifier une date au format Unix et en temps universel coordonné (UTC). La commande suivante permet d'obtenir le statut de toutes les opérations soumises après 00h00 UTC le 1er janvier 2020.

```
aws route53domains list-operations \
    --submitted-since 1577836800
```

- Pour plus de détails sur l'API, reportez-vous ListOperationsà la section Référence des AWS CLI commandes.

`list-tags-for-domain`

L'exemple de code suivant montre comment utiliser`list-tags-for-domain`.

AWS CLI

Pour répertorier les balises d'un domaine

La `list-tags-for-domain` commande suivante répertorie les balises actuellement associées au domaine spécifié.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains list-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": "value2"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources Amazon Route 53 dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListTagsForDomainà la section Référence des AWS CLI commandes.

`register-domain`

L'exemple de code suivant montre comment utiliser`register-domain`.

AWS CLI

Pour enregistrer un domaine

La `register-domain` commande suivante enregistre un domaine et récupère toutes les valeurs de paramètres d'un fichier au format JSON.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains register-domain \
    --region us-east-1 \
    --cli-input-json file://register-domain.json
```

Contenu de `register-domain.json` :

```
{
    "DomainName": "example.com",
    "DurationInYears": 1,
    "AutoRenew": true,
    "AdminContact": {
        "FirstName": "Martha",
        "LastName": "Rivera",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mrivera@example.com"
    },
    "RegistrantContact": {
        "FirstName": "Li",
        "LastName": "Juan",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
```

```
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "ljuan@example.com"
    },
    "TechContact": {
        "FirstName": "Mateo",
        "LastName": "Jackson",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mjackson@example.com"
    },
    "PrivacyProtectAdminContact": true,
    "PrivacyProtectRegistrantContact": true,
    "PrivacyProtectTechContact": true
}
```

Sortie :

```
{
    "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation-detail`. Pour plus d'informations, consultez get-operation-detail.

Pour de plus amples informations, veuillez consulter Enregistrement d'un nouveau domaine dans le Manuel du développeur Amazon Route 53.

Pour plus d'informations sur les domaines de premier niveau (TLDs) nécessitant des valeurs `ExtraParams` et sur les valeurs valides, consultez le manuel ExtraParamde référence des API Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous RegisterDomainà la section Référence des AWS CLI commandes.

`renew-domain`

L'exemple de code suivant montre comment utiliser`renew-domain`.

AWS CLI

Pour renouveler un domaine

La `renew-domain` commande suivante renouvelle le domaine spécifié pour cinq ans. Pour obtenir la valeur de`current-expiry-year`, utilisez la `get-domain-detail` commande et convertissez la valeur `ExpirationDate` de au format Unix.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains renew-domain \
    --region us-east-1 \
    --domain-name example.com \
    --duration-in-years 5 \
    --current-expiry-year 2020
```

Sortie :

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation-detail`. Pour de plus amples informations, veuillez consulter get-operation-detail.

Le registre de chaque domaine de premier niveau (TLD), tel que .com ou .org, contrôle le nombre maximum d'années pour lesquelles vous pouvez renouveler un domaine. Pour connaître la période de renouvellement maximale pour votre domaine, consultez la section « Période d'enregistrement et de renouvellement » de votre TLD dans les domaines que vous pouvez enregistrer auprès d'Amazon Route 53 dans le guide du développeur Amazon Route 53.

Pour plus d'informations, consultez Renouveler l'enregistrement d'un domaine dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous RenewDomainà la section Référence des AWS CLI commandes.

`resend-contact-reachability-email`

L'exemple de code suivant montre comment utiliser`resend-contact-reachability-email`.

AWS CLI

Pour renvoyer l'e-mail de confirmation à l'adresse e-mail actuelle du titulaire, contactez

La `resend-contact-reachability-email` commande suivante renvoie l'e-mail de confirmation à l'adresse e-mail actuelle du contact du titulaire pour le domaine exemple.com.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains resend-contact-reachability-email \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "domainName": "example.com",
    "emailAddress": "moliveira@example.com",
    "isAlreadyVerified": true
}
```

Si la valeur `isAlreadyVerified` est`true`, comme dans cet exemple, le contact du titulaire a déjà confirmé que l'adresse e-mail spécifiée est joignable.

Pour plus d'informations, consultez la section Renvoyer des e-mails d'autorisation et de confirmation dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ResendContactReachabilityEmailà la section Référence des AWS CLI commandes.

`retrieve-domain-auth-code`

L'exemple de code suivant montre comment utiliser`retrieve-domain-auth-code`.

AWS CLI

Pour obtenir le code d'autorisation d'un domaine afin de pouvoir transférer le domaine vers un autre bureau d'enregistrement

La `retrieve-domain-auth-code` commande suivante permet d'obtenir le code d'autorisation actuel pour le domaine exemple.com. Vous donnez cette valeur à un autre bureau d'enregistrement de domaine lorsque vous souhaitez transférer le domaine vers ce bureau d'enregistrement.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains retrieve-domain-auth-code \
    --region us-east-1 \
    --domain-name example.com
```

Sortie :

```
{
    "AuthCode": ")o!v3dJeXampLe"
}
```

Pour plus d'informations, consultez la section Transfert d'un domaine d'Amazon Route 53 vers un autre bureau d'enregistrement dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous RetrieveDomainAuthCodeà la section Référence des AWS CLI commandes.

`transfer-domain`

L'exemple de code suivant montre comment utiliser`transfer-domain`.

AWS CLI

Pour transférer un domaine vers Amazon Route 53

La `transfer-domain` commande suivante transfère un domaine vers Route 53, avec les paramètres fournis par le fichier au format JSON. `C:\temp\transfer-domain.json`

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains transfer-domain \
    --region us-east-1 \
    --cli-input-json file://C:\temp\transfer-domain.json
```

Contenu de `transfer-domain.json` :

```
{
    "DomainName": "example.com",
    "DurationInYears": 1,
    "Nameservers": [
        {
            "Name": "ns-2048.awsdns-64.com"
        },
        {
            "Name": "ns-2049.awsdns-65.net"
        },
        {
            "Name": "ns-2050.awsdns-66.org"
        },
        {
            "Name": "ns-2051.awsdns-67.co.uk"
        }
    ],
    "AuthCode": ")o!v3dJeXampLe",
    "AutoRenew": true,
    "AdminContact": {
        "FirstName": "Martha",
        "LastName": "Rivera",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mrivera@example.com"
    },
    "RegistrantContact": {
        "FirstName": "Li",
        "LastName": "Juan",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
```

```
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "ljuan@example.com"
    },
    "TechContact": {
        "FirstName": "Mateo",
        "LastName": "Jackson",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mjackson@example.com"
    },
    "PrivacyProtectAdminContact": true,
    "PrivacyProtectRegistrantContact": true,
    "PrivacyProtectTechContact": true
}
```

Sortie :

```
{
    "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation-detail`. Pour plus d'informations, consultez get-operation-detail.

Pour plus d'informations, consultez la section Transfert de l'enregistrement d'un domaine vers Amazon Route 53 dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous TransferDomainà la section Référence des AWS CLI commandes.

`update-domain-contact-privacy`

L'exemple de code suivant montre comment utiliser`update-domain-contact-privacy`.

AWS CLI

Pour mettre à jour les paramètres de confidentialité des contacts d'un domaine

La `update-domain-contact-privacy` commande suivante désactive la protection de la confidentialité pour le contact administratif du domaine exemple.com. Cette commande s'exécute uniquement dans la `us-east-1` région.

Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains update-domain-contact-privacy \
    --region us-east-1 \
    --domain-name example.com \
    --no-admin-privacy
```

Sortie :

```
{
    "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter`get-operation-detail`. Pour plus d'informations, consultez get-operation-detail.

Pour plus d'informations, consultez la section Activation ou désactivation de la protection de la confidentialité pour obtenir les coordonnées d'un domaine dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainContactPrivacyà la section Référence des AWS CLI commandes.

`update-domain-contact`

L'exemple de code suivant montre comment utiliser`update-domain-contact`.

AWS CLI

Pour mettre à jour les informations de contact d'un domaine

La `update-domain-contact` commande suivante met à jour les informations de contact d'un domaine en obtenant les paramètres à partir du fichier au format JSON. `C:\temp\update-domain-contact.json`

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains update-domain-contact \
    --region us-east-1 \
    --cli-input-json file://C:\temp\update-domain-contact.json
```

Contenu de `update-domain-contact.json` :

```
{
    "AdminContact": {
        "AddressLine1": "101 Main Street",
        "AddressLine2": "Suite 1a",
        "City": "Seattle",
        "ContactType": "COMPANY",
        "CountryCode": "US",
        "Email": "w.xiulan@example.com",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "OrganizationName": "Example",
        "PhoneNumber": "+1.8005551212",
        "State": "WA",
        "ZipCode": "98101"
    },
    "DomainName": "example.com",
    "RegistrantContact": {
        "AddressLine1": "101 Main Street",
        "AddressLine2": "Suite 1a",
        "City": "Seattle",
        "ContactType": "COMPANY",
        "CountryCode": "US",
        "Email": "w.xiulan@example.com",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "OrganizationName": "Example",
        "PhoneNumber": "+1.8005551212",
        "State": "WA",
        "ZipCode": "98101"
    },
```

```
    "TechContact": {
        "AddressLine1": "101 Main Street",
        "AddressLine2": "Suite 1a",
        "City": "Seattle",
        "ContactType": "COMPANY",
        "CountryCode": "US",
        "Email": "w.xiulan@example.com",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "OrganizationName": "Example",
        "PhoneNumber": "+1.8005551212",
        "State": "WA",
        "ZipCode": "98101"
    }
}
```

Sortie :

```
{
    "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter get-domain-detail. Pour plus d'informations, consultez la section Mise à jour des coordonnées d'un domaine dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainContactà la section Référence des AWS CLI commandes.

`update-domain-nameservers`

L'exemple de code suivant montre comment utiliser`update-domain-nameservers`.

AWS CLI

Pour mettre à jour les serveurs de noms d'un domaine

La `update-domain-nameservers` commande suivante met à jour les serveurs de noms d'un domaine.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains update-domain-nameservers \
    --region us-east-1 \
    --domain-name example.com \
    --
nameservers Name=ns-1.awsdns-01.org Name=ns-2.awsdns-02.co.uk Name=ns-3.awsdns-03.net Name=n
```

Sortie :

```
{
    "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

Pour confirmer que l'opération a réussi, vous pouvez exécuter get-domain-detail.

Pour plus d'informations, consultez la section Ajouter ou modifier des serveurs de noms et des enregistrements Glue pour un domaine dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateDomainNameserversà la section Référence des AWS CLI commandes.

`update-tags-for-domain`

L'exemple de code suivant montre comment utiliser`update-tags-for-domain`.

AWS CLI

Pour ajouter ou mettre à jour des balises pour un domaine

La `update-tags-for-domain` commande suivante ajoute ou met à jour deux clés et les valeurs correspondantes pour le domaine exemple.com. Pour mettre à jour la valeur d'une clé, il suffit d'inclure la clé et la nouvelle valeur. Vous ne pouvez ajouter ou mettre à jour des balises que dans un seul domaine à la fois.

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains update-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com \
    --tags-to-update "Key=key1,Value=value1" "Key=key2,Value=value2"
```

Cette commande ne produit aucun résultat. Pour vérifier que les balises ont été ajoutées ou mises à jour, vous pouvez exécuter list-tags-for-domain.

Pour plus d'informations, consultez la section Marquage des ressources Amazon Route 53 dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous UpdateTagsForDomainà la section Référence des AWS CLI commandes.

`view-billing`

L'exemple de code suivant montre comment utiliser`view-billing`.

AWS CLI

Pour obtenir les informations de facturation relatives aux frais d'enregistrement de domaine pour le AWS compte courant

La `view-billing` commande suivante renvoie tous les enregistrements de facturation relatifs au domaine pour le compte courant pour la période comprise entre le 1er janvier 2018 (1514764800 en heure Unix) et minuit le 31 décembre 2019 (1577836800 en heure Unix).

Cette commande s'exécute uniquement dans la `us-east-1` région. Si votre région par défaut est définie sur`us-east-1`, vous pouvez omettre le `region` paramètre.

```
aws route53domains view-billing \
    --region us-east-1 \
    --start-time 1514764800 \
    --end-time 1577836800
```

Sortie :

```
{
    "BillingRecords": [
        {
            "DomainName": "example.com",
            "Operation": "RENEW_DOMAIN",
            "InvoiceId": "149962827",
            "BillDate": 1536618063.181,
            "Price": 12.0
        },
```

```
        {
            "DomainName": "example.com",
            "Operation": "RENEW_DOMAIN",
            "InvoiceId": "290913289",
            "BillDate": 1568162630.884,
            "Price": 12.0
        }
    ]
}
```

Pour plus d'informations, consultez ViewBillingle manuel de référence des API Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ViewBillingà la section Référence des AWS CLI commandes.

Exemples de profils Route 53 utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide des profils Route 53.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-profile`

L'exemple de code suivant montre comment utiliser`associate-profile`.

AWS CLI

Pour associer un profil

L'`associate-profile`exemple suivant associe un profil à un VPC.

```
aws route53profiles associate-profile \
    --name test-association \
    --profile-id rp-4987774726example \
    --resource-id vpc-0af3b96b3example
```

Sortie :

```
{
    "ProfileAssociation": {
        "CreationTime": 1710851336.527,
        "Id": "rpassoc-489ce212fexample",
        "ModificationTime": 1710851336.527,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "CREATING",
        "StatusMessage": "Creating Profile Association"
    }
}
```

Pour plus d'informations, consultez la section Utilisation des profils dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous AssociateProfileà la section Référence des AWS CLI commandes.

`associate-resource-to-profile`

L'exemple de code suivant montre comment utiliser`associate-resource-to-profile`.

AWS CLI

Pour associer une ressource à un profil

L'`associate-resource-to-profile`exemple suivant associe un groupe de règles de pare-feu DNS ayant la priorité 102 à un profil.

```
aws route53profiles associate-resource-to-profile \
    --name test-resource-association \
    --profile-id rp-4987774726example \
```

```
    --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-
group/rslvr-frg-cfe7f72example \
    --resource-properties "{\"priority\": 102}"
```

Sortie :

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710851216.613,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":102}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Profile to DNS Firewall rule group
 association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateResourceToProfileà la section Référence des AWS CLI commandes.

`create-profile`

L'exemple de code suivant montre comment utiliser`create-profile`.

AWS CLI

Pour créer un profil

L'`create-profile`exemple suivant crée un profil.

```
aws route53profiles create-profile \
    --name test
```

Sortie :

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
        "ClientToken": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": 1710850903.578,
        "Id": "rp-6ffe47d5example",
        "ModificationTime": 1710850903.578,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateProfileà la section Référence des AWS CLI commandes.

`delete-profile`

L'exemple de code suivant montre comment utiliser`delete-profile`.

AWS CLI

Pour supprimer un profil

L'`delete-profile`exemple suivant supprime un profil.

```
aws route53profiles delete-profile \
    --profile-id rp-6ffe47d5example
```

Sortie :

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
        "ClientToken": "0a15fec0-05d9-4f78-bec0-EXAMPLE11111",
        "CreationTime": 1710850903.578,
        "Id": "rp-6ffe47d5example",
```

```
        "ModificationTime": 1710850903.578,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "DELETED",
        "StatusMessage": "Deleted Profile"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteProfileà la section Référence des AWS CLI commandes.

`disassociate-profile`

L'exemple de code suivant montre comment utiliser`disassociate-profile`.

AWS CLI

Pour dissocier un profil

L'`disassociate-profile`exemple suivant dissocie un profil d'un VPC.

```
aws route53profiles disassociate-profile \
    --profile-id rp-4987774726example \
    --resource-id vpc-0af3b96b3example
```

Sortie :

```
{
    "ProfileAssociation": {
        "CreationTime": 1710851336.527,
        "Id": "rpassoc-489ce212fexample",
        "ModificationTime": 1710851401.362,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "DELETING",
        "StatusMessage": "Deleting Profile Association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateProfileà la section Référence des AWS CLI commandes.

`disassociate-resource-from-profile`

L'exemple de code suivant montre comment utiliser`disassociate-resource-from-profile`.

AWS CLI

Pour dissocier une ressource du profil

L'`disassociate-resource-from-profile`exemple suivant dissocie un groupe de règles de pare-feu DNS d'un profil.

```
aws route53profiles disassociate-resource-from-profile \
    --profile-id rp-4987774726example \
    --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example
```

Sortie :

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852624.36,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "DELETING",
        "StatusMessage": "Deleting the Profile to DNS Firewall rule group association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateResourceFromProfileà la section Référence des AWS CLI commandes.

`get-profile-association`

L'exemple de code suivant montre comment utiliser`get-profile-association`.

AWS CLI

Pour obtenir des informations sur une association Profile

Ce qui suit `get-profile-association` renvoie des informations sur l'association de profil spécifiée.

```
aws route53profiles get-profile-association \
    --profile-association-id rpassoc-489ce212fexample
```

Sortie :

```
{
    "ProfileAssociation": {
        "CreationTime": 1709338817.148,
        "Id": "rrpassoc-489ce212fexample",
        "ModificationTime": 1709338974.772,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile Association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetProfileAssociationà la section Référence des AWS CLI commandes.

`get-profile-resource-association`

L'exemple de code suivant montre comment utiliser`get-profile-resource-association`.

AWS CLI

Pour obtenir des informations sur une ressource associée à un profil

Ce qui suit `get-profile-resource-association` renvoie des informations sur l'association de ressources spécifiée à un profil.

```
aws route53profiles get-profile-resource-association \
    --profile-resource-association-id rpr-001913120a7example
```

Sortie :

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852303.798,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "COMPLETE",
        "StatusMessage": "Completed creation of Profile to DNS Firewall rule group
 association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetProfileResourceAssociationà la section Référence des AWS CLI commandes.

`get-profile`

L'exemple de code suivant montre comment utiliser`get-profile`.

AWS CLI

Pour obtenir des informations sur un profil

Ce qui suit `get-profile` renvoie des informations sur le profil spécifié.

```
aws route53profiles get-profile \
    --profile-id rp-4987774726example
```

Sortie :

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
        "ClientToken": "0cbc5ae7-4921-4204-bea9-EXAMPLE11111",
        "CreationTime": 1710851044.288,
        "Id": "rp-4987774726example",
        "ModificationTime": 1710851044.288,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetProfileà la section Référence des AWS CLI commandes.

`list-profile-associations`

L'exemple de code suivant montre comment utiliser`list-profile-associations`.

AWS CLI

Pour répertorier les associations de profils

Vous trouverez ci-dessous la `list-profile-associations` liste des associations de profils de votre AWS compte.

```
aws route53profiles list-profile-associations
```

Sortie :

```
{
    "ProfileAssociations": [
        {
            "CreationTime": 1709338817.148,
            "Id": "rpassoc-489ce212fexample",
```

```
            "ModificationTime": 1709338974.772,
            "Name": "test-association",
            "OwnerId": "123456789012",
            "ProfileId": "rp-4987774726example",
            "ResourceId": "vpc-0af3b96b3example",
            "Status": "COMPLETE",
            "StatusMessage": "Created Profile Association"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListProfileAssociationsà la section Référence des AWS CLI commandes.

`list-profile-resource-associations`

L'exemple de code suivant montre comment utiliser`list-profile-resource-associations`.

AWS CLI

Pour répertorier les associations de ressources du profil

La `list-profile-resource-associations` liste suivante répertorie les associations de ressources de profil pour le profil spécifié.

```
aws route53profiles list-profile-resource-associations \
    --profile-id rp-4987774726example
```

Sortie :

```
{
     "ProfileResourceAssociations": [
         {
             "CreationTime": 1710851216.613,
             "Id": "rpr-001913120a7example",
             "ModificationTime": 1710851216.613,
             "Name": "test-resource-association",
             "OwnerId": "123456789012",
             "ProfileId": "rp-4987774726example",
             "ResourceArn": "arn:aws:route53resolver:us-
east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example",
```

```
            "ResourceProperties": "{\"priority\":102}",
            "ResourceType": "FIREWALL_RULE_GROUP",
            "Status": "COMPLETE",
            "StatusMessage": "Completed creation of Profile to DNS Firewall rule
 group association"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListProfileResourceAssociationsà la section Référence des AWS CLI commandes.

`list-profiles`

L'exemple de code suivant montre comment utiliser`list-profiles`.

AWS CLI

Pour répertorier les profils

Vous trouverez ci-dessous la `list-profiles` liste des profils de votre AWS compte et des informations supplémentaires à leur sujet.

```
aws route53profiles list-profiles
```

Sortie :

```
{
    "ProfileSummaries": [
        {
            "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
            "Id": "rp-4987774726example",
            "Name": "test",
            "ShareStatus": "NOT_SHARED"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListProfilesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

La `list-tags-for-resource` liste suivante répertorie les balises associées à la ressource spécifiée.

```
aws route53profiles list-tags-for-resource \
    --resource-arn arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example
```

Sortie :

```
{
    "Tags": {
        "my-key-2": "my-value-2",
        "my-key-1": "my-value-1"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`update-profile-resource-association`

L'exemple de code suivant montre comment utiliserupdate-profile-resource-association.

AWS CLI

Pour mettre à jour une ressource associée à un profil

Ce qui suit `update-profile-resource-association` met à jour la priorité d'un groupe de règles de pare-feu DNS associé au profil.

```
aws route53profiles update-profile-resource-association \
    --profile-resource-association-id rpr-001913120a7example \
```

```
    --resource-properties "{\"priority\": 105}"
```

Sortie :

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852303.798,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Profile to DNS Firewall rule group
 association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateProfileResourceAssociationà la section Référence des AWS CLI commandes.

Exemples de résolveurs Route 53 utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du résolveur AWS Command Line Interface with Route 53.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-firewall-rule-group`

L'exemple de code suivant montre comment utiliser`associate-firewall-rule-group`.

AWS CLI

Pour associer un groupe de règles de pare-feu à un VPC

L'`associate-firewall-rule-group`exemple suivant associe un groupe de règles de pare-feu DNS à un Amazon VPC.

```
aws route53resolver associate-firewall-rule-group \
    --name test-association \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --vpc-id vpc-31e92222 \
    --priority 101
```

Sortie :

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 101,
        "MutationProtection": "DISABLED",
        "Status": "UPDATING",
        "StatusMessage": "Creating Firewall Rule Group Association",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:47:48.755768Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des associations entre votre VPC et les groupes de règles du pare-feu DNS Route 53 Resolver dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous AssociateFirewallRuleGroupà la section Référence des AWS CLI commandes.

`associate-resolver-endpoint-ip-address`

L'exemple de code suivant montre comment utiliser`associate-resolver-endpoint-ip-address`.

AWS CLI

Pour associer une autre adresse IP à un point de terminaison du résolveur

L'`associate-resolver-endpoint-ip-address`exemple suivant associe une autre adresse IP à un point de terminaison Resolver entrant. Si vous spécifiez uniquement un ID de sous-réseau et que vous omettez l'adresse IP dans le `--ip-address` paramètre, Resolver choisit une adresse IP pour vous parmi les adresses IP disponibles dans le sous-réseau spécifié.

```
aws route53resolver associate-resolver-endpoint-ip-address \
    --resolver-endpoint-id rslvr-in-497098ad5example \
    --ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"
```

Sortie :

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-497098ad5example",
        "CreatorRequestId": "AWSConsole.25.0123456789",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad5example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 3,
        "HostVPCId": "vpc-304bexam",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Resolver Endpoint",
        "CreationTime": "2020-01-02T23:25:45.538Z",
        "ModificationTime": "2020-01-02T23:25:45.538Z"
    }
```

```
}
```

Pour plus d'informations, consultez la section Valeurs que vous spécifiez lorsque vous créez ou modifiez des points de terminaison entrants dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous AssociateResolverEndpointIpAddressà la section Référence des AWS CLI commandes.

`associate-resolver-rule`

L'exemple de code suivant montre comment utiliser`associate-resolver-rule`.

AWS CLI

Pour associer une règle de résolution à un VPC

L'`associate-resolver-rule`exemple suivant associe une règle Resolver à un Amazon VPC. Après avoir exécuté la commande, Resolver commence à transférer les requêtes DNS vers votre réseau en fonction des paramètres de la règle, tels que le nom de domaine des requêtes transférées.

```
aws route53resolver associate-resolver-rule \
    --name my-resolver-rule-association \
    --resolver-rule-id rslvr-rr-42b60677c0example \
    --vpc-id vpc-304bexam
```

Sortie :

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-d61cbb2c8bexample",
        "ResolverRuleId": "rslvr-rr-42b60677c0example",
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "CREATING",
        "StatusMessage": "[Trace id: 1-5dc5a8fa-ec2cc480d2ef07617example] Creating
 the association."
    }
}
```

Pour plus d'informations, consultez la section Transfert de requêtes DNS sortantes vers votre réseau dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateResolverRuleà la section Référence des AWS CLI commandes.

`create-firewall-domain-list`

L'exemple de code suivant montre comment utiliser`create-firewall-domain-list`.

AWS CLI

Pour créer une liste de domaines de pare-feu DNS Route 53 Resolver

L'`create-firewall-domain-list`exemple suivant crée une liste de domaines du pare-feu DNS Route 53 Resolver, nommée test, dans votre AWS compte.

```
aws route53resolver create-firewall-domain-list \
    --creator-request-id my-request-id \
    --name test
```

Sortie :

```
{
    "FirewallDomainList": {
        "Id": "rslvr-fdl-d61cbb2cbexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/
rslvr-fdl-d61cbb2cbexample",
        "Name": "test",
        "DomainCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Domain List",
        "CreatorRequestId": "my-request-id",
        "CreationTime": "2021-05-25T15:55:51.115365Z",
        "ModificationTime": "2021-05-25T15:55:51.115365Z"
    }
}
```

Pour plus d'informations, consultez Gérer vos propres listes de domaines dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateFirewallDomainListà la section Référence des AWS CLI commandes.

`create-firewall-rule-group`

L'exemple de code suivant montre comment utiliser`create-firewall-rule-group`.

AWS CLI

Pour créer un groupe de règles de pare-feu

L'`create-firewall-rule-group`exemple suivant crée un groupe de règles de pare-feu DNS.

```
aws route53resolver create-firewall-rule-group \
    --creator-request-id my-request-id \
    --name test
```

Sortie :

```
{
    "FirewallRuleGroup": {
        "Id": "rslvr-frg-47f93271fexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
        "Name": "test",
        "RuleCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Rule Group",
        "OwnerId": "123456789012",
        "CreatorRequestId": "my-request-id",
        "ShareStatus": "NOT_SHARED",
        "CreationTime": "2021-05-25T18:59:26.490017Z",
        "ModificationTime": "2021-05-25T18:59:26.490017Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateFirewallRuleGroupà la section Référence des AWS CLI commandes.

`create-firewall-rule`

L'exemple de code suivant montre comment utiliser`create-firewall-rule`.

AWS CLI

Pour créer une règle de pare-feu

L'`create-firewall-rule`exemple suivant crée une règle de pare-feu dans une règle de pare-feu DNS pour les domaines répertoriés dans une liste de domaines de pare-feu DNS.

```
aws route53resolver create-firewall-rule \
    --name allow-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
    --priority 101 \
    --action ALLOW
```

Sortie :

```
{
    "FirewallRule": {
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
        "Name": "allow-rule",
        "Priority": 101,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:44:00.346093Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateFirewallRuleà la section Référence des AWS CLI commandes.

`create-resolver-endpoint`

L'exemple de code suivant montre comment utiliser`create-resolver-endpoint`.

AWS CLI

Pour créer un point de terminaison Resolver entrant

L'`create-resolver-endpoint`exemple suivant crée un point de terminaison Resolver entrant. Vous pouvez utiliser la même commande pour créer des points de terminaison entrants et sortants.

aws route53resolver create-resolver-endpoint --name — creator-request-id 2020-01-18:47 — security-group-ids « sg-f62bexam » my-inbound-endpoint —direction INBOUND --ip-addresses =sous-net-ba47exam, Ip=192.0.2.255 =sous-net-12d8exam, Ip=192.0.2.254 SubnetId SubnetId

Sortie :

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-f9ab8a03f1example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-f9ab8a03f1example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-f62bexam"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304examp",
        "Status": "CREATING",
        "StatusMessage": "[Trace id: 1-5dc1ff84-f3477826e4a190025example] Creating
 the Resolver Endpoint",
        "CreationTime": "2020-01-01T23:02:29.583Z",
        "ModificationTime": "2020-01-01T23:02:29.583Z"
    }
}
```

Pour créer un point de terminaison Resolver sortant

L'`create-resolver-endpoint`exemple suivant crée un point de terminaison de résolution sortant à l'aide des valeurs du document au format JSON. `create-outbound-resolver-endpoint.json`

```
aws route53resolver create-resolver-endpoint \
    --cli-input-json file://c:\temp\create-outbound-resolver-endpoint.json
```

Contenu de `create-outbound-resolver-endpoint.json` :

```
{
   "CreatorRequestId": "2020-01-01-18:47",
   "Direction": "OUTBOUND",
   "IpAddresses": [
      {
         "Ip": "192.0.2.255",
         "SubnetId": "subnet-ba47exam"
      },
      {
         "Ip": "192.0.2.254",
         "SubnetId": "subnet-12d8exam"
      }
   ],
   "Name": "my-outbound-endpoint",
   "SecurityGroupIds": [ "sg-05cd7b25d6example" ],
   "Tags": [
      {
         "Key": "my-key-name",
         "Value": "my-key-value"
      }
   ]
}
```

Pour plus d'informations, consultez la section Résolution des requêtes DNS entre VPCs et votre réseau dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateResolverEndpointà la section Référence des AWS CLI commandes.

`create-resolver-rule`

L'exemple de code suivant montre comment utiliser`create-resolver-rule`.

AWS CLI

Pour créer une règle de résolution

L'`create-resolver-rule`exemple suivant crée une règle de transfert Resolver. La règle utilise le point de terminaison sortant rslvr-out-d 5e5920e37example pour transférer les requêtes DNS vers les adresses IP 10.24.8.75 et `example.com` 10.24.8.156.

```
aws route53resolver create-resolver-rule \
```

```
    --creator-request-id 2020-01-02-18:47 \
    --domain-name example.com \
    --name my-rule \
    --resolver-endpoint-id rslvr-out-d5e5920e37example \
    --rule-type FORWARD \
    --target-ips "Ip=10.24.8.75" "Ip=10.24.8.156"
```

Sortie :

```
{
    "ResolverRule": {
        "Status": "COMPLETE",
        "RuleType": "FORWARD",
        "ResolverEndpointId": "rslvr-out-d5e5920e37example",
        "Name": "my-rule",
        "DomainName": "example.com.",
        "CreationTime": "2022-05-10T21:35:30.923187Z",
        "TargetIps": [
            {
                "Ip": "10.24.8.75",
                "Port": 53
            },
            {
                "Ip": "10.24.8.156",
                "Port": 53
            }
        ],
        "CreatorRequestId": "2022-05-10-16:33",
        "ModificationTime": "2022-05-10T21:35:30.923187Z",
        "ShareStatus": "NOT_SHARED",
        "Arn": "arn:aws:route53resolver:us-east-1:111117012054:resolver-rule/rslvr-
rr-b1e0b905e93611111",
        "OwnerId": "111111111111",
        "Id": "rslvr-rr-rslvr-rr-b1e0b905e93611111",
        "StatusMessage": "[Trace id: 1-22222222-3e56afcc71a3724664f22e24]
 Successfully created Resolver Rule."
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateResolverRuleà la section Référence des AWS CLI commandes.

`delete-firewall-domain-list`

L'exemple de code suivant montre comment utiliser`delete-firewall-domain-list`.

AWS CLI

Pour supprimer une liste de domaines du pare-feu DNS Route 53 Resolver

L'`delete-firewall-domain-list`exemple suivant supprime une liste de domaines du pare-feu DNS Route 53 Resolver, nommée test, de votre AWS compte.

```
aws route53resolver delete-firewall-domain-list \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Sortie :

```
{
    "FirewallDomainList": {
        "Id": "rslvr-fdl-9e956e9ffexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/
rslvr-fdl-9e956e9ffexample",
        "Name": "test",
        "DomainCount": 6,
        "Status": "DELETING",
        "StatusMessage": "Deleting the Firewall Domain List",
        "CreatorRequestId": "my-request-id",
        "CreationTime": "2021-05-25T15:55:51.115365Z",
        "ModificationTime": "2021-05-25T18:58:05.588024Z"
    }
}
```

Pour plus d'informations, consultez Gérer vos propres listes de domaines dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteFirewallDomainListà la section Référence des AWS CLI commandes.

`delete-firewall-rule-group`

L'exemple de code suivant montre comment utiliser`delete-firewall-rule-group`.

AWS CLI

Pour supprimer un groupe de règles de pare-feu

L'`delete-firewall-rule-group`exemple suivant supprime un groupe de règles de pare-feu.

```
aws route53resolver delete-firewall-rule-group \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Sortie :

```
{
    "FirewallRuleGroup": {
        "Id": "rslvr-frg-47f93271fexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
        "Name": "test",
        "RuleCount": 0,
        "Status": "UPDATING",
        "StatusMessage": "Updating Firewall Rule Group",
        "OwnerId": "123456789012",
        "CreatorRequestId": "my-request-id",
        "ShareStatus": "NOT_SHARED",
        "CreationTime": "2021-05-25T18:59:26.490017Z",
        "ModificationTime": "2021-05-25T21:51:53.028688Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteFirewallRuleGroupà la section Référence des AWS CLI commandes.

`delete-firewall-rule`

L'exemple de code suivant montre comment utiliser`delete-firewall-rule`.

AWS CLI

Pour supprimer une règle de pare-feu

L'`delete-firewall-rule`exemple suivant supprime une règle de pare-feu spécifiée.

```
aws route53resolver delete-firewall-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Sortie :

```
{
    "FirewallRule": {
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
        "Name": "allow-rule",
        "Priority": 102,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:45:59.611600Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteFirewallRuleà la section Référence des AWS CLI commandes.

`delete-resolver-endpoint`

L'exemple de code suivant montre comment utiliser`delete-resolver-endpoint`.

AWS CLI

Pour supprimer un point de terminaison du résolveur

L'`delete-resolver-endpoint`exemple suivant supprime le point de terminaison spécifié.

Important Si vous supprimez un point de terminaison entrant, les requêtes DNS de votre réseau ne sont plus transmises à Resolver dans le VPC que vous avez spécifié dans le point de terminaison. Si vous supprimez un point de terminaison sortant, Resolver cessera de réacheminer les requêtes DNS à partir de votre VPC vers votre réseau pour les règles qui spécifient le point de terminaison sortant supprimé.

```
aws route53resolver delete-resolver-endpoint \
    --resolver-endpoint-id rslvr-in-497098ad59example
```

Sortie :

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-497098ad59example",
        "CreatorRequestId": "AWSConsole.25.157290example",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad59example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 5,
        "HostVPCId": "vpc-304bexam",
        "Status": "DELETING",
        "StatusMessage": "[Trace id: 1-5dc5b658-811b5be0922bbc382example] Deleting
 ResolverEndpoint.",
        "CreationTime": "2020-01-01T23:25:45.538Z",
        "ModificationTime": "2020-01-02T23:25:45.538Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteResolverEndpointà la section Référence des AWS CLI commandes.

`delete-resolver-rule`

L'exemple de code suivant montre comment utiliser`delete-resolver-rule`.

AWS CLI

Pour supprimer une règle de résolution

L'`delete-resolver-rule`exemple suivant supprime la règle spécifiée.

Remarque Si une règle est associée à une règle VPCs, vous devez d'abord la dissocier VPCs avant de pouvoir la supprimer.

```
aws route53resolver delete-resolver-rule \
    --resolver-rule-id rslvr-rr-5b3809426bexample
```

Sortie :

```
{
    "ResolverRule": {
        "Id": "rslvr-rr-5b3809426bexample",
        "CreatorRequestId": "2020-01-03-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-
rr-5b3809426bexample",
        "DomainName": "zenith.example.com.",
        "Status": "DELETING",
        "StatusMessage": "[Trace id: 1-5dc5e05b-602e67b052cb74f05example] Deleting
 Resolver Rule.",
        "RuleType": "FORWARD",
        "Name": "my-resolver-rule",
        "TargetIps": [
            {
                "Ip": "192.0.2.50",
                "Port": 53
            }
        ],
        "ResolverEndpointId": "rslvr-out-d5e5920e3example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteResolverRuleà la section Référence des AWS CLI commandes.

`disassociate-firewall-rule-group`

L'exemple de code suivant montre comment utiliser`disassociate-firewall-rule-group`.

AWS CLI

Pour dissocier un groupe de règles de pare-feu d'un VPC

L'`disassociate-firewall-rule-group`exemple suivant dissocie un groupe de règles de pare-feu DNS d'un Amazon VPC.

```
aws route53resolver disassociate-firewall-rule-group \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

Sortie :

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 103,
        "MutationProtection": "DISABLED",
        "Status": "DELETING",
        "StatusMessage": "Deleting the Firewall Rule Group Association",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:51:02.377887Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des associations entre votre VPC et les groupes de règles du pare-feu DNS Route 53 Resolver dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous DisassociateFirewallRuleGroupà la section Référence des AWS CLI commandes.

`disassociate-resolver-endpoint-ip-address`

L'exemple de code suivant montre comment utiliser`disassociate-resolver-endpoint-ip-address`.

AWS CLI

Pour dissocier une adresse IP d'un point de terminaison du résolveur

L'`disassociate-resolver-endpoint-ip-address`exemple suivant supprime une adresse IP d'un point de terminaison entrant ou sortant du Resolver spécifié.

Remarque Un point de terminaison doit avoir au moins deux adresses IP. Si un point de terminaison ne possède actuellement que deux adresses IP et que vous souhaitez remplacer une adresse par une autre, vous devez d'abord utiliser associate-resolver-endpoint-ip-address pour associer la nouvelle adresse IP. Vous pouvez ensuite dissocier l'une des adresses IP d'origine du point de terminaison.

```
aws route53resolver disassociate-resolver-endpoint-ip-address \
    --resolver-endpoint-id rslvr-in-f9ab8a03f1example \
    --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"
```

Sortie :

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-f9ab8a03f1example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-f9ab8a03f1example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-f62bexam"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 3,
        "HostVPCId": "vpc-304bexam",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Resolver Endpoint",
        "CreationTime": "2020-01-01T23:02:29.583Z",
        "ModificationTime": "2020-01-05T23:02:29.583Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateResolverEndpointIpAddressà la section Référence des AWS CLI commandes.

`disassociate-resolver-rule`

L'exemple de code suivant montre comment utiliser`disassociate-resolver-rule`.

AWS CLI

Pour dissocier une règle de résolution d'un Amazon VPC

L'`disassociate-resolver-rule`exemple suivant supprime l'association entre la règle de résolution spécifiée et le VPC spécifié. Vous pouvez dissocier une règle d'un VPC dans les cas suivants :

Pour les requêtes DNS provenant de ce VPC, vous souhaitez que Resolver arrête de transférer les requêtes vers votre réseau pour le nom de domaine spécifié dans la règle. Vous souhaitez supprimer la règle de transfert. Si une règle est actuellement associée à une ou plusieurs d'entre VPCs elles, vous devez la dissocier de toutes les règles VPCs avant de pouvoir la supprimer.

```
aws route53resolver disassociate-resolver-rule \
    --resolver-rule-id rslvr-rr-4955cb98ceexample \
    --vpc-id vpc-304bexam
```

Sortie :

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-322f4e8b9cexample",
        "ResolverRuleId": "rslvr-rr-4955cb98ceexample",
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "DELETING",
        "StatusMessage": "[Trace id: 1-5dc5ffa2-a26c38004c1f94006example] Deleting
 Association"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateResolverRuleà la section Référence des AWS CLI commandes.

`get-firewall-config`

L'exemple de code suivant montre comment utiliser`get-firewall-config`.

AWS CLI

Pour obtenir une configuration de pare-feu pour un VPC

L'`get-firewall-config`exemple suivant récupère le comportement du pare-feu DNS pour le VPC spécifié.

```
aws route53resolver get-firewall-config \
    --resource-id vpc-31e92222
```

Sortie :

```
{
    "FirewallConfig": {
        "Id": "rslvr-fc-86016850cexample",
        "ResourceId": "vpc-31e9222",
        "OwnerId": "123456789012",
        "FirewallFailOpen": "DISABLED"
    }
}
```

Pour plus d'informations, consultez la section Configuration VPC du pare-feu DNS dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous GetFirewallConfigà la section Référence des AWS CLI commandes.

`get-firewall-domain-list`

L'exemple de code suivant montre comment utiliser`get-firewall-domain-list`.

AWS CLI

Pour obtenir une liste de domaines du pare-feu DNS Route 53 Resolver

L'`get-firewall-domain-list`exemple suivant récupère la liste des domaines avec l'ID que vous spécifiez.

```
aws route53resolver get-firewall-domain-list \
    --firewall-domain-list-id rslvr-fdl-42b60677cexample
```

Sortie :

```
{
```

```
    "FirewallDomainList": {
        "Id": "rslvr-fdl-9e956e9ffexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123457689012:firewall-domain-list/
rslvr-fdl-42b60677cexample",
        "Name": "test",
        "DomainCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Domain List",
        "CreatorRequestId": "my-request-id",
        "CreationTime": "2021-05-25T15:55:51.115365Z",
        "ModificationTime": "2021-05-25T15:55:51.115365Z"
    }
}
```

Pour plus d'informations, consultez Gérer vos propres listes de domaines dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetFirewallDomainListà la section Référence des AWS CLI commandes.

`get-firewall-rule-group-association`

L'exemple de code suivant montre comment utiliser`get-firewall-rule-group-association`.

AWS CLI

Pour obtenir une association de groupes de règles de pare-feu

L'`get-firewall-rule-group-association`exemple suivant récupère une association de groupes de règles de pare-feu.

```
aws route53resolver get-firewall-rule-group-association \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

Sortie :

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
```

```
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 101,
        "MutationProtection": "DISABLED",
        "Status": "COMPLETE",
        "StatusMessage": "Finished rule group association update",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:47:48.755768Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des associations entre votre VPC et les groupes de règles du pare-feu DNS Route 53 Resolver dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous GetFirewallRuleGroupAssociationà la section Référence des AWS CLI commandes.

`get-firewall-rule-group-policy`

L'exemple de code suivant montre comment utiliser`get-firewall-rule-group-policy`.

AWS CLI

Pour obtenir une politique AWS IAM

L'`get-firewall-rule-group-policy`exemple suivant obtient la politique AWS Identity and Access Management (AWS IAM) pour partager le groupe de règles spécifié.

```
aws route53resolver get-firewall-rule-group-policy \
    --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/
rslvr-frg-47f93271fexample
```

Sortie :

```
{
    "FirewallRuleGroupPolicy": "{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Sid\":\"test\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"arn:aws:iam::AWS_ACCOUNT_ID:root\"},\"Action\":
[\"route53resolver:GetFirewallRuleGroup\",\"route53resolver:ListFirewallRuleGroups
```

```
\"],\"Resource\":\"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-group/rslvr-frg-47f93271fexample\"}]}"
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetFirewallRuleGroupPolicyà la section Référence des AWS CLI commandes.

`get-firewall-rule-group`

L'exemple de code suivant montre comment utiliser`get-firewall-rule-group`.

AWS CLI

Pour obtenir un groupe de règles de pare-feu

L'`get-firewall-rule-group`exemple suivant permet de récupérer des informations sur un groupe de règles de pare-feu DNS avec l'ID que vous fournissez.

```
aws route53resolver get-firewall-rule-group \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Sortie :

```
{
    "FirewallRuleGroup": {
        "Id": "rslvr-frg-47f93271fexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/rslvr-frg-47f93271fexample",
        "Name": "test",
        "RuleCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Rule Group",
        "OwnerId": "123456789012",
        "CreatorRequestId": "my-request-id",
        "ShareStatus": "NOT_SHARED",
        "CreationTime": "2021-05-25T18:59:26.490017Z",
        "ModificationTime": "2021-05-25T18:59:26.490017Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetFirewallRuleGroupà la section Référence des AWS CLI commandes.

`get-resolver-endpoint`

L'exemple de code suivant montre comment utiliser`get-resolver-endpoint`.

AWS CLI

Pour obtenir des informations sur un point de terminaison Resolver

L'`get-resolver-endpoint`exemple suivant affiche les détails du point de terminaison sortant spécifié. Vous pouvez l'utiliser `get-resolver-endpoint` pour les points de terminaison entrants et sortants en spécifiant l'ID de point de terminaison applicable.

```
aws route53resolver get-resolver-endpoint \
    --resolver-endpoint-id rslvr-out-d5e5920e37example
```

Sortie :

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-out-d5e5920e37example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-out-d5e5920e37example",
        "Name": "my-outbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "OUTBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304bexam",
        "Status": "OPERATIONAL",
        "StatusMessage": "This Resolver Endpoint is operational.",
        "CreationTime": "2020-01-01T23:50:50.979Z",
        "ModificationTime": "2020-01-02T23:50:50.979Z"
    }
}
```

Pour plus d'informations, consultez la section Valeurs que vous spécifiez lorsque vous créez ou modifiez des points de terminaison entrants dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous GetResolverEndpointà la section Référence des AWS CLI commandes.

`get-resolver-rule-association`

L'exemple de code suivant montre comment utiliser`get-resolver-rule-association`.

AWS CLI

Pour obtenir des informations sur l'association entre une règle de résolution et un VPC

L'`get-resolver-rule-association`exemple suivant affiche des détails sur l'association entre une règle de résolution spécifiée et un VPC. Vous associez une règle de résolution et un associate-resolver-ruleVPC à l'aide de.

```
aws route53resolver get-resolver-rule-association \
    --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample
```

Sortie :

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-d61cbb2c8bexample",
        "ResolverRuleId": "rslvr-rr-42b60677c0example",
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "COMPLETE",
        "StatusMessage": ""
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetResolverRuleAssociationà la section Référence des AWS CLI commandes.

`get-resolver-rule`

L'exemple de code suivant montre comment utiliser`get-resolver-rule`.

AWS CLI

Pour obtenir des informations sur une règle du résolveur

L'`get-resolver-rule`exemple suivant affiche des détails sur la règle de résolution spécifiée, tels que le nom de domaine pour lequel la règle transmet les requêtes DNS et l'ID du point de terminaison du résolveur sortant auquel la règle est associée.

```
aws route53resolver get-resolver-rule \
    --resolver-rule-id rslvr-rr-42b60677c0example
```

Sortie :

```
{
    "ResolverRule": {
        "Id": "rslvr-rr-42b60677c0example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-
rr-42b60677c0example",
        "DomainName": "example.com.",
        "Status": "COMPLETE",
        "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example]
 Successfully created Resolver Rule.",
        "RuleType": "FORWARD",
        "Name": "my-rule",
        "TargetIps": [
            {
                "Ip": "192.0.2.45",
                "Port": 53
            }
        ],
        "ResolverEndpointId": "rslvr-out-d5e5920e37example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
}
```

Pour plus d'informations, consultez la section Valeurs que vous spécifiez lorsque vous créez ou modifiez des règles dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous GetResolverRuleà la section Référence des AWS CLI commandes.

`import-firewall-domains`

L'exemple de code suivant montre comment utiliser`import-firewall-domains`.

AWS CLI

Pour importer des domaines dans une liste de domaines

L'`import-firewall-domains`exemple suivant importe un ensemble de domaines d'un fichier dans une liste de domaines du pare-feu DNS que vous spécifiez.

```
aws route53resolver import-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample \
    --operation REPLACE \
    --domain-file-url s3://PATH/TO/YOUR/FILE
```

Sortie :

```
{
    "Id": "rslvr-fdl-d61cbb2cbexample",
    "Name": "test",
    "Status": "IMPORTING",
    "StatusMessage": "Importing domains from provided file."
}
```

Pour plus d'informations, consultez Gérer vos propres listes de domaines dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ImportFirewallDomainsà la section Référence des AWS CLI commandes.

`list-firewall-configs`

L'exemple de code suivant montre comment utiliser`list-firewall-configs`.

AWS CLI

Pour répertorier les configurations de pare-feu

L'`list-firewall-configs`exemple suivant répertorie les configurations de votre pare-feu DNS.

```
aws route53resolver list-firewall-configs
```

Sortie :

```
{
    "FirewallConfigs": [
        {
            "Id": "rslvr-fc-86016850cexample",
            "ResourceId": "vpc-31e92222",
            "OwnerId": "123456789012",
            "FirewallFailOpen": "DISABLED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration VPC du pare-feu DNS dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListFirewallConfigsà la section Référence des AWS CLI commandes.

`list-firewall-domain-lists`

L'exemple de code suivant montre comment utiliser`list-firewall-domain-lists`.

AWS CLI

Pour répertorier toutes les listes de domaines du pare-feu DNS Route 53 Resolver

L'`list-firewall-domain-lists`exemple suivant répertorie toutes les listes de domaines.

```
aws route53resolver list-firewall-domain-lists
```

Sortie :

```
{
    "FirewallDomainLists": [
        {
            "Id": "rslvr-fdl-2c46f2ecfexample",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-
list/rslvr-fdl-2c46f2ecfexample",
            "Name": "AWSManagedDomainsMalwareDomainList",
```

```
            "CreatorRequestId": "AWSManagedDomainsMalwareDomainList",
            "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
        },
        {
            "Id": "rslvr-fdl-aa970e9e1example",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-
list/rslvr-fdl-aa970e9e1example",
            "Name": "AWSManagedDomainsBotnetCommandandControl",
            "CreatorRequestId": "AWSManagedDomainsBotnetCommandandControl",
            "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
        },
        {
            "Id": "rslvr-fdl-42b60677cexample",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789111:firewall-domain-
list/rslvr-fdl-42b60677cexample",
            "Name": "test",
            "CreatorRequestId": "my-request-id"
        }
    ]
}
```

Pour plus d'informations, consultez les listes de domaines du pare-feu DNS Route 53 Resolver dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListFirewallDomainListsà la section Référence des AWS CLI commandes.

`list-firewall-domains`

L'exemple de code suivant montre comment utiliser`list-firewall-domains`.

AWS CLI

Pour répertorier des domaines dans une liste de domaines

L'`list-firewall-domains`exemple suivant répertorie les domaines d'une liste de domaines de pare-feu DNS que vous spécifiez.

```
aws route53resolver list-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample
```

Sortie :

```
{
    "Domains": [
        "test1.com.",
        "test2.com.",
        "test3.com."
    ]
}
```

Pour plus d'informations, consultez Gérer vos propres listes de domaines dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListFirewallDomainsà la section Référence des AWS CLI commandes.

`list-firewall-rule-group-associations`

L'exemple de code suivant montre comment utiliser`list-firewall-rule-group-associations`.

AWS CLI

Pour répertorier les associations de groupes de règles du pare-feu DNS

L'`list-firewall-rule-group-associations`exemple suivant répertorie vos associations de groupes de règles de pare-feu DNS avec Amazon VPCs.

```
aws route53resolver list-firewall-rule-group-associations
```

Sortie :

```
{
    "FirewallRuleGroupAssociations": [
        {
            "Id": "rslvr-frgassoc-57e8873d7example",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-
group-association/rslvr-frgassoc-57e8873d7example",
            "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
            "VpcId": "vpc-31e92222",
            "Name": "test-association",
            "Priority": 101,
            "MutationProtection": "DISABLED",
```

```
            "Status": "UPDATING",
            "StatusMessage": "Creating Firewall Rule Group Association",
            "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
            "CreationTime": "2021-05-25T21:47:48.755768Z",
            "ModificationTime": "2021-05-25T21:47:48.755768Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des associations entre votre VPC et le groupe de règles du pare-feu DNS Route 53 Resolver dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListFirewallRuleGroupAssociationsà la section Référence des AWS CLI commandes.

`list-firewall-rule-groups`

L'exemple de code suivant montre comment utiliser`list-firewall-rule-groups`.

AWS CLI

Pour obtenir la liste de vos groupes de règles de pare-feu

L'`list-firewall-rule-groups`exemple suivant répertorie vos groupes de règles de pare-feu DNS.

```
aws route53resolver list-firewall-rule-groups
```

Sortie :

```
{
    "FirewallRuleGroups": [
        {
            "Id": "rslvr-frg-47f93271fexample",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-
group/rslvr-frg-47f93271fexample",
            "Name": "test",
            "OwnerId": "123456789012",
            "CreatorRequestId": "my-request-id",
            "ShareStatus": "NOT_SHARED"
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListFirewallRuleGroupsà la section Référence des AWS CLI commandes.

`list-firewall-rules`

L'exemple de code suivant montre comment utiliser`list-firewall-rules`.

AWS CLI

Pour répertorier les règles de pare-feu

L'`list-firewall-rules`exemple suivant répertorie toutes vos règles de pare-feu DNS au sein d'un groupe de règles de pare-feu.

```
aws route53resolver list-firewall-rules \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Sortie :

```
{
    "FirewallRules": [
        {
            "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
            "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
            "Name": "allow-rule",
            "Priority": 101,
            "Action": "ALLOW",
            "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
            "CreationTime": "2021-05-25T21:44:00.346093Z",
            "ModificationTime": "2021-05-25T21:44:00.346093Z"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListFirewallRulesà la section Référence des AWS CLI commandes.

`list-resolver-endpoint-ip-addresses`

L'exemple de code suivant montre comment utiliser`list-resolver-endpoint-ip-addresses`.

AWS CLI

Pour répertorier les adresses IP d'un point de terminaison entrant ou sortant spécifié

L'`list-resolver-endpoint-ip-addresses`exemple suivant répertorie les informations relatives aux adresses IP associées au point de terminaison `rslvr-in-f9ab8a03f1example` entrant. Vous pouvez également l'utiliser `list-resolver-endpoint-ip-addresses` pour les points de terminaison sortants en spécifiant l'ID de point de terminaison applicable.

```
aws route53resolver list-resolver-endpoint-ip-addresses \
    --resolver-endpoint-id rslvr-in-f9ab8a03f1example
```

Sortie :

```
{
    "MaxResults": 10,
    "IpAddresses": [
        {
            "IpId": "rni-1de60cdbfeexample",
            "SubnetId": "subnet-ba47exam",
            "Ip": "192.0.2.44",
            "Status": "ATTACHED",
            "StatusMessage": "This IP address is operational.",
            "CreationTime": "2020-01-03T23:02:29.587Z",
            "ModificationTime": "2020-01-03T23:03:05.555Z"
        },
        {
            "IpId": "rni-aac7085e38example",
            "SubnetId": "subnet-12d8exam",
            "Ip": "192.0.2.45",
            "Status": "ATTACHED",
            "StatusMessage": "This IP address is operational.",
            "CreationTime": "2020-01-03T23:02:29.593Z",
            "ModificationTime": "2020-01-03T23:02:55.060Z"
        }
```

```
    ]
}
```

Pour plus d'informations sur les valeurs de la sortie, consultez les sections Valeurs que vous spécifiez lorsque vous créez ou modifiez des points de terminaison entrants et Valeurs que vous spécifiez lorsque vous créez ou modifiez des points de terminaison sortants, toutes deux dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListResolverEndpointIpAddressesà la section Référence des AWS CLI commandes.

`list-resolver-endpoints`

L'exemple de code suivant montre comment utiliser`list-resolver-endpoints`.

AWS CLI

Pour répertorier les points de terminaison du résolveur dans une région AWS

L'`list-resolver-endpoints`exemple suivant répertorie les points de terminaison Resolver entrants et sortants qui existent dans le compte courant.

```
aws route53resolver list-resolver-endpoints
```

Sortie :

```
{
    "MaxResults": 10,
    "ResolverEndpoints": [
        {
            "Id": "rslvr-in-497098ad59example",
            "CreatorRequestId": "2020-01-01-18:47",
            "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-in-497098ad59example",
            "Name": "my-inbound-endpoint",
            "SecurityGroupIds": [
                "sg-05cd7b25d6example"
            ],
            "Direction": "INBOUND",
            "IpAddressCount": 2,
            "HostVPCId": "vpc-304bexam",
```

```
            "Status": "OPERATIONAL",
            "StatusMessage": "This Resolver Endpoint is operational.",
            "CreationTime": "2020-01-01T23:25:45.538Z",
            "ModificationTime": "2020-01-01T23:25:45.538Z"
        },
        {
            "Id": "rslvr-out-d5e5920e37example",
            "CreatorRequestId": "2020-01-01-18:48",
            "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-out-d5e5920e37example",
            "Name": "my-outbound-endpoint",
            "SecurityGroupIds": [
                "sg-05cd7b25d6example"
            ],
            "Direction": "OUTBOUND",
            "IpAddressCount": 2,
            "HostVPCId": "vpc-304bexam",
            "Status": "OPERATIONAL",
            "StatusMessage": "This Resolver Endpoint is operational.",
            "CreationTime": "2020-01-01T23:50:50.979Z",
            "ModificationTime": "2020-01-01T23:50:50.979Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResolverEndpointsà la section Référence des AWS CLI commandes.

`list-resolver-rule-associations`

L'exemple de code suivant montre comment utiliser`list-resolver-rule-associations`.

AWS CLI

Pour répertorier les associations entre les règles du résolveur et VPCs

L'`list-resolver-rule-associations`exemple suivant répertorie les associations entre les règles du résolveur et VPCs celles du AWS compte courant.

```
aws route53resolver list-resolver-rule-associations
```

Sortie :

```
{
    "MaxResults": 30,
    "ResolverRuleAssociations": [
        {
            "Id": "rslvr-autodefined-assoc-vpc-304bexam-internet-resolver",
            "ResolverRuleId": "rslvr-autodefined-rr-internet-resolver",
            "Name": "System Rule Association",
            "VPCId": "vpc-304bexam",
            "Status": "COMPLETE",
            "StatusMessage": ""
        },
        {
            "Id": "rslvr-rrassoc-d61cbb2c8bexample",
            "ResolverRuleId": "rslvr-rr-42b60677c0example",
            "Name": "my-resolver-rule-association",
            "VPCId": "vpc-304bexam",
            "Status": "COMPLETE",
            "StatusMessage": ""
        }
    ]
}
```

Pour plus d'informations, consultez la section Comment le résolveur Route 53 transmet les requêtes DNS de votre part VPCs à votre réseau dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListResolverRuleAssociationsà la section Référence des AWS CLI commandes.

`list-resolver-rules`

L'exemple de code suivant montre comment utiliser`list-resolver-rules`.

AWS CLI

Pour répertorier les règles du résolveur

L'`list-resolver-rules`exemple suivant répertorie toutes les règles Resolver du AWS compte courant.

```
aws route53resolver list-resolver-rules
```

Sortie :

```
{
    "MaxResults": 30,
    "ResolverRules": [
        {
            "Id": "rslvr-autodefined-rr-internet-resolver",
            "CreatorRequestId": "",
            "Arn": "arn:aws:route53resolver:us-west-2::autodefined-rule/rslvr-
autodefined-rr-internet-resolver",
            "DomainName": ".",
            "Status": "COMPLETE",
            "RuleType": "RECURSIVE",
            "Name": "Internet Resolver",
            "OwnerId": "Route 53 Resolver",
            "ShareStatus": "NOT_SHARED"
        },
        {
            "Id": "rslvr-rr-42b60677c0example",
            "CreatorRequestId": "2020-01-01-18:47",
            "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0bc4e299",
            "DomainName": "example.com.",
            "Status": "COMPLETE",
            "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example]
 Successfully created Resolver Rule.",
            "RuleType": "FORWARD",
            "Name": "my-rule",
            "TargetIps": [
                {
                    "Ip": "192.0.2.45",
                    "Port": 53
                }
            ],
            "ResolverEndpointId": "rslvr-out-d5e5920e37example",
            "OwnerId": "111122223333",
            "ShareStatus": "NOT_SHARED"
        }
    ]
}
```

Pour plus d'informations, consultez la section Comment le résolveur Route 53 transmet les requêtes DNS de votre part VPCs à votre réseau dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous ListResolverRulesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource Resolver

L'`list-tags-for-resource`exemple suivant répertorie les balises attribuées à la règle Resolver spécifiée.

```
aws route53resolver list-tags-for-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example"
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "my-key-1",
            "Value": "my-value-1"
        },
        {
            "Key": "my-key-2",
            "Value": "my-value-2"
        }
    ]
}
```

Pour plus d'informations sur l'utilisation des balises pour la répartition des coûts, consultez la section Using Cost Allocation Tags du guide de l'utilisateur de AWS Billing and Cost Management.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`put-firewall-rule-group-policy`

L'exemple de code suivant montre comment utiliser`put-firewall-rule-group-policy`.

AWS CLI

Pour associer une stratégie AWS IAM afin de partager une stratégie de groupe de règles de pare-feu

L'`put-firewall-rule-group-policy`exemple suivant joint une politique AWS Identity and Access Management (AWS IAM) pour partager le groupe de règles.

```
aws route53resolver put-firewall-rule-group-policy \
    --firewall-rule-group-policy "{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Sid\":\"test\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"arn:aws:iam::AWS_ACCOUNT_ID:root\"},\"Action\":
[\"route53resolver:GetFirewallRuleGroup\",\"route53resolver:ListFirewallRuleGroups
\"],\"Resource\":\"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-
group/rslvr-frg-47f93271fexample\"}]}"
```

Sortie :

```
{
    "ReturnValue": true
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutFirewallRuleGroupPolicyà la section Référence des AWS CLI commandes.

`put-resolver-rule-policy`

L'exemple de code suivant montre comment utiliser`put-resolver-rule-policy`.

AWS CLI

Pour partager une règle Resolver avec un autre compte AWS

L'`put-resolver-rule-policy`exemple suivant indique une règle de résolution que vous souhaitez partager avec un autre AWS compte, le compte avec lequel vous souhaitez partager la

règle et les opérations liées aux règles que vous souhaitez que le compte puisse effectuer sur ces règles.

Remarque Vous devez exécuter cette commande à l'aide des informations d'identification du compte qui a créé la règle.

```
aws route53resolver put-resolver-rule-policy \
    --region us-east-1 \
    --arn "arn:aws:route53resolver:us-east-1:111122223333:resolver-rule/rslvr-rr-42b60677c0example" \
    --resolver-rule-policy "{\"Version\": \"2012-10-17\", \
        \"Statement\": [ { \
        \"Effect\" : \"Allow\", \
        \"Principal\" : {\"AWS\" : \"444455556666\" }, \
        \"Action\" : [ \
            \"route53resolver:GetResolverRule\", \
            \"route53resolver:AssociateResolverRule\", \
            \"route53resolver:DisassociateResolverRule\", \
            \"route53resolver:ListResolverRules\", \
            \"route53resolver:ListResolverRuleAssociations\" ], \
        \"Resource\" : [ \"arn:aws:route53resolver:us-east-1:111122223333:resolver-rule/rslvr-rr-42b60677c0example\" ] } ] }"
```

Sortie :

```
{
    "ReturnValue": true
}
```

Après l'exécution`put-resolver-rule-policy`, vous pouvez exécuter les deux commandes Resource Access Manager (RAM) suivantes. Vous devez utiliser le compte avec lequel vous souhaitez partager la règle :

`get-resource-share-invitations`renvoie la valeur`resourceShareInvitationArn`. Vous avez besoin de cette valeur pour accepter l'invitation à utiliser la règle partagée. `accept-resource-share-invitation`accepte l'invitation à utiliser la règle partagée.

Pour plus d’informations, consultez la documentation de suivante :

get-resource-share-invitationsaccept-resource-share-invitationsPartage des règles de transfert avec d'autres AWS comptes et utilisation de règles partagées dans le manuel du développeur Amazon Route 53

- Pour plus de détails sur l'API, reportez-vous PutResolverRulePolicyà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour associer des balises à une ressource Resolver

L'`tag-resource`exemple suivant associe deux key/value paires de balises à la règle Resolver spécifiée.

```
aws route53resolver tag-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example" \
    --tags "Key=my-key-1,Value=my-value-1" "Key=my-key-2,Value=my-value-2"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur l'utilisation des balises pour la répartition des coûts, consultez la section Using Cost Allocation Tags du guide de l'utilisateur de AWS Billing and Cost Management.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource de résolution

L'`untag-resource`exemple suivant supprime deux balises de la règle Resolver spécifiée.

```
aws route53resolver untag-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example" \
```

```
    --tag-keys my-key-1 my-key-2
```

Cette commande ne produit aucun résultat. Pour confirmer que les balises ont été supprimées, vous pouvez utiliser list-tags-for-resource.

Pour plus d'informations sur l'utilisation des balises pour la répartition des coûts, consultez la section Using Cost Allocation Tags du guide de l'utilisateur de AWS Billing and Cost Management.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-firewall-config`

L'exemple de code suivant montre comment utiliser`update-firewall-config`.

AWS CLI

Pour mettre à jour la configuration d'un pare-feu

L'`update-firewall-config`exemple suivant met à jour la configuration du pare-feu DNS.

```
aws route53resolver update-firewall-config \
    --resource-id vpc-31e92222 \
    --firewall-fail-open DISABLED
```

Sortie :

```
{
    "FirewallConfig": {
        "Id": "rslvr-fc-86016850cexample",
        "ResourceId": "vpc-31e92222",
        "OwnerId": "123456789012",
        "FirewallFailOpen": "DISABLED"
    }
}
```

Pour plus d'informations, consultez la section Configuration VPC du pare-feu DNS dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous UpdateFirewallConfigà la section Référence des AWS CLI commandes.

`update-firewall-domains`

L'exemple de code suivant montre comment utiliser`update-firewall-domains`.

AWS CLI

Pour mettre à jour une liste de domaines

L'`update-firewall-domains`exemple suivant ajoute les domaines à une liste de domaines avec l'ID que vous fournissez.

```
aws route53resolver update-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-42b60677cexampleb \
    --operation ADD \
    --domains test1.com test2.com test3.com
```

Sortie :

```
{
    "Id": "rslvr-fdl-42b60677cexample",
    "Name": "test",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Firewall Domain List"
}
```

Pour plus d'informations, consultez Gérer vos propres listes de domaines dans le manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateFirewallDomainsà la section Référence des AWS CLI commandes.

`update-firewall-rule-group-association`

L'exemple de code suivant montre comment utiliser`update-firewall-rule-group-association`.

AWS CLI

Pour mettre à jour une association de groupes de règles de pare-feu

L'`update-firewall-rule-group-association`exemple suivant met à jour une association de groupes de règles de pare-feu.

```
aws route53resolver update-firewall-rule-group-association \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \
    --priority 103
```

Sortie :

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 103,
        "MutationProtection": "DISABLED",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Firewall Rule Group Association Attributes",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:50:09.272569Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des associations entre votre VPC et le groupe de règles du pare-feu DNS Route 53 Resolver dans le manuel du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous UpdateFirewallRuleGroupAssociationà la section Référence des AWS CLI commandes.

`update-firewall-rule`

L'exemple de code suivant montre comment utiliser`update-firewall-rule`.

AWS CLI

Pour mettre à jour une règle de pare-feu

L'`update-firewall-rule`exemple suivant met à jour une règle de pare-feu avec les paramètres que vous spécifiez.

```
aws route53resolver update-firewall-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
    --priority 102
```

Sortie :

```
{
    "FirewallRule": {
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
        "Name": "allow-rule",
        "Priority": 102,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:45:59.611600Z"
    }
}
```

Pour plus d'informations, consultez la section Gestion des groupes de règles et des règles dans le pare-feu DNS du manuel Amazon Route 53 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateFirewallRuleà la section Référence des AWS CLI commandes.

`update-resolver-endpoint`

L'exemple de code suivant montre comment utiliser`update-resolver-endpoint`.

AWS CLI

Pour mettre à jour le nom d'un point de terminaison du résolveur

L'`update-resolver-endpoint`exemple suivant met à jour le nom d'un point de terminaison du résolveur. La mise à jour d'autres valeurs n'est pas prise en charge.

```
aws route53resolver update-resolver-endpoint \
    --resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \
    --name my-renamed-inbound-endpoint
```

Sortie :

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-b5d45e32bdexample",
        "CreatorRequestId": "2020-01-02-18:48",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-b5d45e32bdexample",
        "Name": "my-renamed-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-f62bexam"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304bexam",
        "Status": "OPERATIONAL",
        "StatusMessage": "This Resolver Endpoint is operational.",
        "CreationTime": "2020-01-01T18:33:59.265Z",
        "ModificationTime": "2020-01-08T18:33:59.265Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateResolverEndpointà la section Référence des AWS CLI commandes.

`update-resolver-rule`

L'exemple de code suivant montre comment utiliser`update-resolver-rule`.

AWS CLI

Exemple 1 : pour mettre à jour les paramètres du point de terminaison du résolveur

L'`update-resolver-rule`exemple suivant met à jour le nom de la règle, les adresses IP de votre réseau local vers lesquelles les requêtes DNS sont transférées et l'ID du point de terminaison du résolveur sortant que vous utilisez pour transférer les requêtes vers votre réseau.

Remarque Les valeurs existantes pour `TargetIps` sont remplacées. Vous devez donc spécifier toutes les adresses IP que vous souhaitez attribuer à la règle après la mise à jour.

```
aws route53resolver update-resolver-rule \
```

```
    --resolver-rule-id rslvr-rr-1247fa64f3example \
    --config Name="my-2nd-rule",TargetIps=[{Ip=192.0.2.45,Port=53},
{Ip=192.0.2.46,Port=53}],ResolverEndpointId=rslvr-out-7b89ed0d25example
```

Sortie :

```
{
    "ResolverRule": {
        "Id": "rslvr-rr-1247fa64f3example",
        "CreatorRequestId": "2020-01-02-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-
rr-1247fa64f3example",
        "DomainName": "www.example.com.",
        "Status": "COMPLETE",
        "StatusMessage": "[Trace id: 1-5dcc90b9-8a8ee860aba1ebd89example]
 Successfully updated Resolver Rule.",
        "RuleType": "FORWARD",
        "Name": "my-2nd-rule",
        "TargetIps": [
            {
                "Ip": "192.0.2.45",
                "Port": 53
            },
            {
                "Ip": "192.0.2.46",
                "Port": 53
            }
        ],
        "ResolverEndpointId": "rslvr-out-7b89ed0d25example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
}
```

Exemple 2 : Pour mettre à jour les paramètres du point de terminaison du résolveur à l'aide d'un fichier pour les paramètres ``config``

Vous pouvez également inclure les `config` paramètres dans un fichier JSON, puis spécifier ce fichier lors de l'appel`update-resolver-rule`.

```
aws route53resolver update-resolver-rule \
    --resolver-rule-id rslvr-rr-1247fa64f3example \
```

```
    --config file://c:\temp\update-resolver-rule.json
```

Contenu de `update-resolver-rule.json`.

```
{
    "Name": "my-2nd-rule",
    "TargetIps": [
        {
            "Ip": "192.0.2.45",
            "Port": 53
        },
        {
            "Ip": "192.0.2.46",
            "Port": 53
        }
    ],
    "ResolverEndpointId": "rslvr-out-7b89ed0d25example"
}
```

Pour plus d'informations, consultez la section Valeurs que vous spécifiez lorsque vous créez ou modifiez des règles dans le guide du développeur Amazon Route 53.

- Pour plus de détails sur l'API, reportez-vous UpdateResolverRuleà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon S3 AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon S3.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`abort-multipart-upload`

L'exemple de code suivant montre comment utiliser`abort-multipart-upload`.

AWS CLI

Pour annuler le téléchargement partitionné spécifié

La `abort-multipart-upload` commande suivante interrompt le téléchargement partitionné de la clé `multipart/01` dans le compartiment. `amzn-s3-demo-bucket`

```
aws s3api abort-multipart-upload \
    --bucket amzn-s3-demo-bucket \
    --key multipart/01 \
    --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

L'ID de téléchargement requis par cette commande est généré par `create-multipart-upload` et peut également être récupéré avec`list-multipart-uploads`.

- Pour plus de détails sur l'API, reportez-vous AbortMultipartUploadà la section Référence des AWS CLI commandes.

`complete-multipart-upload`

L'exemple de code suivant montre comment utiliser`complete-multipart-upload`.

AWS CLI

La commande suivante effectue un téléchargement en plusieurs parties pour la clé contenue `multipart/01` dans le compartiment `amzn-s3-demo-bucket` :

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket amzn-s3-demo-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

L'ID de téléchargement requis par cette commande est généré par `create-multipart-upload` et peut également être récupéré avec`list-multipart-uploads`.

L'option de téléchargement partitionné de la commande ci-dessus utilise une structure JSON qui décrit les parties du téléchargement partitionné qui doivent être réassemblées dans le fichier complet. Dans cet exemple, le préfixe `file://` est utilisé pour charger la structure JSON à partir d'un fichier du dossier local nommé `mpustruct`.

structure :

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

La ETag valeur de chaque partie à télécharger est affichée chaque fois que vous téléchargez une pièce à l'aide de la `upload-part` commande et peut également être récupérée en appelant `list-parts` ou calculée en prenant la MD5 somme de contrôle de chaque partie.

Sortie :

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
    "Bucket": "amzn-s3-demo-bucket",
    "Location": "https://amzn-s3-demo-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- Pour plus de détails sur l'API, reportez-vous CompleteMultipartUploadà la section Référence des AWS CLI commandes.

`copy-object`

L'exemple de code suivant montre comment utiliser`copy-object`.

AWS CLI

La commande suivante copie un objet de `bucket-1` vers `bucket-2` :

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2
```

Sortie :

```
{
    "CopyObjectResult": {
        "LastModified": "2015-11-10T01:07:25.000Z",
        "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
    },
    "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS CLI commandes.

`cp`

L'exemple de code suivant montre comment utiliser`cp`.

AWS CLI

Exemple 1 : Copier un fichier local vers S3

La `cp` commande suivante copie un seul fichier dans un compartiment et une clé spécifiés :

```
aws s3 cp test.txt s3://amzn-s3-demo-bucket/test2.txt
```

Sortie :

```
upload: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 2 : Copier un fichier local vers S3 avec une date d'expiration

La cp commande suivante copie un seul fichier dans un compartiment et une clé spécifiés qui expirent à l'horodatage ISO 8601 spécifié :

```
aws s3 cp test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --expires 2014-10-01T20:30:00Z
```

Sortie :

```
upload: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 3 : Copier un fichier de S3 vers S3

La cp commande suivante copie un seul objet s3 dans un compartiment et une clé spécifiés :

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt
```

Sortie :

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 4 : Copier un objet S3 dans un fichier local

La cp commande suivante copie localement un seul objet dans un fichier spécifié :

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt test2.txt
```

Sortie :

```
download: s3://amzn-s3-demo-bucket/test.txt to test2.txt
```

Exemple 5 : Copier un objet S3 d'un compartiment à un autre

La cp commande suivante copie un seul objet dans un compartiment spécifié tout en conservant son nom d'origine :

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket2/
```

Sortie :

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

Exemple 6 : copie récursive d'objets S3 dans un répertoire local

Lorsqu'elle est passée avec le paramètre`--recursive`, la `cp` commande suivante copie de manière récursive tous les objets sous un préfixe et un compartiment spécifiés dans un répertoire spécifié. Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et `test2.txt` :

```
aws s3 cp s3://amzn-s3-demo-bucket . \
    --recursive
```

Sortie :

```
download: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
download: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

Exemple 7 : Copie récursive de fichiers locaux vers S3

Lorsqu'elle est passée avec le paramètre`--recursive`, la `cp` commande suivante copie de manière récursive tous les fichiers d'un répertoire spécifié vers un compartiment et un préfixe spécifiés, tout en excluant certains fichiers à l'aide d'un `--exclude` paramètre. Dans cet exemple, le répertoire `myDir` contient les fichiers `test1.txt` et `test2.jpg` :

```
aws s3 cp myDir s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

Sortie :

```
upload: myDir/test1.txt to s3://amzn-s3-demo-bucket/test1.txt
```

Exemple 8 : copie récursive d'objets S3 vers un autre compartiment

Lorsqu'elle est passée avec le paramètre`--recursive`, la `cp` commande suivante copie de manière récursive tous les objets d'un compartiment spécifié vers un autre compartiment tout en excluant certains objets à l'aide d'un `--exclude` paramètre. Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et `another/test1.txt` :

```
aws s3 cp s3://amzn-s3-demo-bucket/ s3://amzn-s3-demo-bucket2/ \
    --recursive \
    --exclude "another/*"
```

Sortie :

```
copy: s3://amzn-s3-demo-bucket/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

Vous pouvez combiner `--exclude` des `--include` options pour ne copier que les objets correspondant à un modèle, à l'exclusion de tous les autres :

```
aws s3 cp s3://amzn-s3-demo-bucket/logs/ s3://amzn-s3-demo-bucket2/logs/ \
    --recursive \
    --exclude "*" \
    --include "*.log"
```

Sortie :

```
copy: s3://amzn-s3-demo-bucket/logs/test/test.log to s3://amzn-s3-demo-bucket2/logs/
test/test.log
copy: s3://amzn-s3-demo-bucket/logs/test3.log to s3://amzn-s3-demo-bucket2/logs/
test3.log
```

Exemple 9 : Configuration de la liste de contrôle d'accès (ACL) lors de la copie d'un objet S3

La `cp` commande suivante copie un seul objet dans un compartiment et une clé spécifiés tout en définissant l'ACL sur `public-read-write` :

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --acl public-read-write
```

Sortie :

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Notez que si vous utilisez cette `--acl` option, assurez-vous que toutes les politiques IAM associées incluent l'`"s3:PutObjectAcl"`action :

```
aws iam get-user-policy \
    --user-name myuser \
    --policy-name mypolicy
```

Sortie :

```
{
    "UserName": "myuser",
    "PolicyName": "mypolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "s3:PutObject",
                    "s3:PutObjectAcl"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                ],
                "Effect": "Allow",
                "Sid": "Stmt1234567891234"
            }
        ]
    }
}
```

Exemple 10 : Octroi d'autorisations pour un objet S3

La `cp` commande suivante illustre l'utilisation de l'`--grants`option permettant d'accorder l'accès en lecture à tous les utilisateurs identifiés par l'URI et le contrôle total à un utilisateur spécifique identifié par son identifiant canonique :

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://
acs.amazonaws.com/groups/global/
AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Sortie :

```
upload: file.txt to s3://amzn-s3-demo-bucket/file.txt
```

Exemple 11 : Chargement d'un flux de fichiers local vers S3

PowerShell peut modifier le codage ou ajouter un CRLF à une entrée redirigée.

La `cp` commande suivante télécharge un flux de fichiers local depuis l'entrée standard vers un compartiment et une clé spécifiés :

```
aws s3 cp - s3://amzn-s3-demo-bucket/stream.txt
```

Exemple 12 : Chargement d'un flux de fichiers local de plus de 50 Go vers S3

La `cp` commande suivante télécharge un flux de fichiers local de 51 Go depuis l'entrée standard vers un compartiment et une clé spécifiés. L'`--expected-size`option doit être fournie, sinon le téléchargement risque d'échouer lorsqu'il atteint la limite de 10 000 parties par défaut :

```
aws s3 cp - s3://amzn-s3-demo-bucket/stream.txt --expected-size 54760833024
```

Exemple 13 : Téléchargement d'un objet S3 en tant que flux de fichiers local

PowerShell peut modifier le codage ou ajouter un CRLF à une sortie redirigée ou redirigée.

La `cp` commande suivante télécharge un objet S3 localement sous forme de flux vers la sortie standard. Le téléchargement sous forme de flux n'est actuellement pas compatible avec le `--recursive` paramètre :

```
aws s3 cp s3://amzn-s3-demo-bucket/stream.txt -
```

Exemple 14 : Téléchargement vers un point d'accès S3

La `cp` commande suivante télécharge un seul fichier (`mydoc.txt`) vers le point d'accès (`myaccesspoint`) au niveau de la touche (`mykey`) :

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

Sortie :

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

Exemple 15 : Téléchargement depuis un point d'accès S3

La `cp` commande suivante télécharge un seul objet (`mykey`) du point d'accès (`myaccesspoint`) vers le fichier local (`mydoc.txt`) :

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey mydoc.txt
```

Sortie :

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to
 mydoc.txt
```

- Pour plus de détails sur l'API, voir Cp dans AWS CLI Command Reference.

`create-bucket`

L'exemple de code suivant montre comment utiliser`create-bucket`.

AWS CLI

Exemple 1 : pour créer un bucket

L'`create-bucket`exemple suivant crée un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region us-east-1
```

Sortie :

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

Pour plus d'informations, consultez la section Création d'un compartiment dans le guide de l'utilisateur Amazon S3.

Exemple 2 : pour créer un bucket avec l'obligation du propriétaire

L'`create-bucket`exemple suivant crée un bucket nommé `amzn-s3-demo-bucket` qui utilise le paramètre imposé par le propriétaire du bucket pour S3 Object Ownership.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region us-east-1 \
    --object-ownership BucketOwnerEnforced
```

Sortie :

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

Pour plus d'informations, consultez la section Contrôle de la propriété des objets et désactivation ACLs dans le guide de l'utilisateur Amazon S3.

Exemple 3 : Pour créer un bucket en dehors de la région ``us-east-1``

L'`create-bucket`exemple suivant crée un compartiment nommé `amzn-s3-demo-bucket` dans la `eu-west-1` région. Les régions situées en dehors de `us-east-1` doivent être spécifiées afin de créer le compartiment dans la région souhaitée. `LocationConstraint`

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

Sortie :

```
{
    "Location": "http://amzn-s3-demo-bucket.s3.amazonaws.com/"
}
```

Pour plus d'informations, consultez la section Création d'un compartiment dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS CLI commandes.

`create-multipart-upload`

L'exemple de code suivant montre comment utiliser`create-multipart-upload`.

AWS CLI

La commande suivante crée un téléchargement partitionné dans le bucket `amzn-s3-demo-bucket` avec la clé `multipart/01` :

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket --key 'multipart/01'
```

Sortie :

```
{
    "Bucket": "amzn-s3-demo-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
    "Key": "multipart/01"
}
```

Le fichier terminé sera nommé `01` dans un dossier appelé `multipart` dans le compartiment`amzn-s3-demo-bucket`. Enregistrez l'ID de téléchargement, la clé et le nom du compartiment à utiliser avec la `upload-part` commande.

- Pour plus de détails sur l'API, reportez-vous CreateMultipartUploadà la section Référence des AWS CLI commandes.

`delete-bucket-analytics-configuration`

L'exemple de code suivant montre comment utiliser`delete-bucket-analytics-configuration`.

AWS CLI

Pour supprimer une configuration d'analyse pour un bucket

L'`delete-bucket-analytics-configuration`exemple suivant supprime la configuration d'analyse pour le bucket et l'ID spécifiés.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

`delete-bucket-cors`

L'exemple de code suivant montre comment utiliser`delete-bucket-cors`.

AWS CLI

La commande suivante supprime une configuration de partage de ressources entre origines d'un compartiment nommé : `amzn-s3-demo-bucket`

```
aws s3api delete-bucket-cors --bucket amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketCorsà la section Référence des AWS CLI commandes.

`delete-bucket-encryption`

L'exemple de code suivant montre comment utiliser`delete-bucket-encryption`.

AWS CLI

Pour supprimer la configuration de chiffrement côté serveur d'un bucket

L'`delete-bucket-encryption`exemple suivant supprime la configuration de chiffrement côté serveur du compartiment spécifié.

```
aws s3api delete-bucket-encryption \
    --bucket amzn-s3-demo-bucket
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketEncryptionà la section Référence des AWS CLI commandes.

`delete-bucket-intelligent-tiering-configuration`

L'exemple de code suivant montre comment utiliser`delete-bucket-intelligent-tiering-configuration`.

AWS CLI

Pour supprimer une configuration S3 Intelligent-Tiering sur un bucket

L'`delete-bucket-intelligent-tiering-configuration`exemple suivant supprime une configuration S3 Intelligent-Tiering, nommée ExampleConfig, sur un bucket.

```
aws s3api delete-bucket-intelligent-tiering-configuration \
```

```
    --bucket amzn-s3-demo-bucket \
    --id ExampleConfig
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Utilisation de S3 Intelligent-Tiering dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketIntelligentTieringConfigurationà la section Référence des AWS CLI commandes.

`delete-bucket-inventory-configuration`

L'exemple de code suivant montre comment utiliser`delete-bucket-inventory-configuration`.

AWS CLI

Pour supprimer la configuration d'inventaire d'un bucket

L'`delete-bucket-inventory-configuration`exemple suivant supprime la configuration d'inventaire avec l'ID 1 du compartiment spécifié.

```
aws s3api delete-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketInventoryConfigurationà la section Référence des AWS CLI commandes.

`delete-bucket-lifecycle`

L'exemple de code suivant montre comment utiliser`delete-bucket-lifecycle`.

AWS CLI

La commande suivante supprime une configuration de cycle de vie d'un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-bucket-lifecycle --bucket amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketLifecycleà la section Référence des AWS CLI commandes.

`delete-bucket-metrics-configuration`

L'exemple de code suivant montre comment utiliser`delete-bucket-metrics-configuration`.

AWS CLI

Pour supprimer la configuration des métriques d'un bucket

L'`delete-bucket-metrics-configuration`exemple suivant supprime la configuration des métriques pour le bucket et l'ID spécifiés.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

`delete-bucket-ownership-controls`

L'exemple de code suivant montre comment utiliser`delete-bucket-ownership-controls`.

AWS CLI

Pour supprimer les paramètres de propriété d'un bucket

L'`delete-bucket-ownership-controls`exemple suivant supprime les paramètres de propriété d'un bucket.

```
aws s3api delete-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition de la propriété d'un objet sur un compartiment existant dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketOwnershipControlsà la section Référence des AWS CLI commandes.

`delete-bucket-policy`

L'exemple de code suivant montre comment utiliser`delete-bucket-policy`.

AWS CLI

La commande suivante supprime une politique de bucket d'un bucket nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-bucket-policy --bucket amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS CLI commandes.

`delete-bucket-replication`

L'exemple de code suivant montre comment utiliser`delete-bucket-replication`.

AWS CLI

La commande suivante supprime une configuration de réplication d'un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-bucket-replication --bucket amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketReplicationà la section Référence des AWS CLI commandes.

`delete-bucket-tagging`

L'exemple de code suivant montre comment utiliser`delete-bucket-tagging`.

AWS CLI

La commande suivante supprime une configuration de balisage d'un compartiment nommé : `amzn-s3-demo-bucket`

```
aws s3api delete-bucket-tagging --bucket amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketTaggingà la section Référence des AWS CLI commandes.

`delete-bucket-website`

L'exemple de code suivant montre comment utiliser`delete-bucket-website`.

AWS CLI

La commande suivante supprime la configuration d'un site Web d'un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-bucket-website --bucket amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketWebsiteà la section Référence des AWS CLI commandes.

`delete-bucket`

L'exemple de code suivant montre comment utiliser`delete-bucket`.

AWS CLI

La commande suivante supprime un bucket nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-bucket --bucket amzn-s3-demo-bucket --region us-east-1
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS CLI commandes.

`delete-object-tagging`

L'exemple de code suivant montre comment utiliser`delete-object-tagging`.

AWS CLI

Pour supprimer les ensembles de balises d'un objet

L'`delete-object-tagging`exemple suivant supprime de l'objet `doc1.rtf` la balise avec la clé spécifiée.

```
aws s3api delete-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteObjectTaggingà la section Référence des AWS CLI commandes.

`delete-object`

L'exemple de code suivant montre comment utiliser`delete-object`.

AWS CLI

La commande suivante supprime un objet nommé `test.txt` dans un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key test.txt
```

Si le versionnement des compartiments est activé, la sortie contiendra l'ID de version du marqueur de suppression :

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
  "DeleteMarker": true
}
```

Pour plus d'informations sur la suppression d'objets, consultez Supprimer des objets dans le manuel Amazon S3 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS CLI commandes.

`delete-objects`

L'exemple de code suivant montre comment utiliser`delete-objects`.

AWS CLI

La commande suivante supprime un objet d'un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete file://delete.json
```

`delete.json`est un document JSON dans le répertoire en cours qui indique l'objet à supprimer :

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Sortie :

```
{
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS CLI commandes.

`delete-public-access-block`

L'exemple de code suivant montre comment utiliser`delete-public-access-block`.

AWS CLI

Pour supprimer la configuration de blocage de l'accès public pour un bucket

L'`delete-public-access-block`exemple suivant supprime la configuration de blocage de l'accès public sur le compartiment spécifié.

```
aws s3api delete-public-access-block \
    --bucket amzn-s3-demo-bucket
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePublicAccessBlockà la section Référence des AWS CLI commandes.

`get-bucket-accelerate-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-accelerate-configuration`.

AWS CLI

Pour récupérer la configuration accélérée d'un bucket

L'`get-bucket-accelerate-configuration`exemple suivant récupère la configuration d'accélération pour le compartiment spécifié.

```
aws s3api get-bucket-accelerate-configuration \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Status": "Enabled"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAccelerateConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-acl`

L'exemple de code suivant montre comment utiliser`get-bucket-acl`.

AWS CLI

La commande suivante permet de récupérer la liste de contrôle d'accès pour un bucket nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAclà la section Référence des AWS CLI commandes.

`get-bucket-analytics-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-analytics-configuration`.

AWS CLI

Pour récupérer la configuration d'analyse d'un bucket avec un ID spécifique

L'`get-bucket-analytics-configuration`exemple suivant affiche la configuration d'analyse pour le bucket et l'ID spécifiés.

```
aws s3api get-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

Sortie :

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-cors`

L'exemple de code suivant montre comment utiliser`get-bucket-cors`.

AWS CLI

La commande suivante extrait la configuration du partage de ressources entre origines pour un compartiment nommé : `amzn-s3-demo-bucket`

```
aws s3api get-bucket-cors --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "CORSRules": [
        {
            "AllowedHeaders": [
                "*"
            ],
            "ExposeHeaders": [
                "x-amz-server-side-encryption"
            ],
            "AllowedMethods": [
                "PUT",
                "POST",
                "DELETE"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                "http://www.example.com"
            ]
```

```
        },
        {
            "AllowedHeaders": [
                "Authorization"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedMethods": [
                "GET"
            ],
            "AllowedOrigins": [
                "*"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketCorsà la section Référence des AWS CLI commandes.

`get-bucket-encryption`

L'exemple de code suivant montre comment utiliser`get-bucket-encryption`.

AWS CLI

Pour récupérer la configuration de chiffrement côté serveur pour un bucket

L'`get-bucket-encryption`exemple suivant récupère la configuration de chiffrement côté serveur pour le compartiment. `amzn-s3-demo-bucket`

```
aws s3api get-bucket-encryption \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
```

```
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketEncryptionà la section Référence des AWS CLI commandes.

`get-bucket-intelligent-tiering-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-intelligent-tiering-configuration`.

AWS CLI

Pour récupérer une configuration S3 Intelligent-Tiering sur un bucket

L'`get-bucket-intelligent-tiering-configuration`exemple suivant récupère une configuration S3 Intelligent-Tiering, nommée ExampleConfig, sur un bucket.

```
aws s3api get-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-bucket \
    --id ExampleConfig
```

Sortie :

```
{
    "IntelligentTieringConfiguration": {
        "Id": "ExampleConfig2",
        "Filter": {
            "Prefix": "images"
        },
        "Status": "Enabled",
        "Tierings": [
            {
                "Days": 90,
                "AccessTier": "ARCHIVE_ACCESS"
            },
            {
                "Days": 180,
                "AccessTier": "DEEP_ARCHIVE_ACCESS"
            }
```

```
        ]
    }
}
```

Pour plus d'informations, consultez la section Utilisation de S3 Intelligent-Tiering dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous GetBucketIntelligentTieringConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-inventory-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-inventory-configuration`.

AWS CLI

Pour récupérer la configuration d'inventaire d'un bucket

L'`get-bucket-inventory-configuration`exemple suivant récupère la configuration d'inventaire pour le compartiment spécifié avec l'ID1.

```
aws s3api get-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

Sortie :

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                "AccountId": "123456789012"
            }
        },
        "IncludedObjectVersions": "Current",
        "Id": "1",
        "Schedule": {
            "Frequency": "Weekly"
        }
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketInventoryConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-lifecycle-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-lifecycle-configuration`.

AWS CLI

La commande suivante permet de récupérer la configuration du cycle de vie d'un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Rules": [
        {
            "ID": "Move rotated logs to Glacier",
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2015-11-10T00:00:00.000Z",
                    "StorageClass": "GLACIER"
                }
            ]
        },
        {
            "Status": "Enabled",
            "Prefix": "",
            "NoncurrentVersionTransitions": [
                {
                    "NoncurrentDays": 0,
                    "StorageClass": "GLACIER"
                }
            ],
            "ID": "Move old versions to Glacier"
```

```
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLifecycleConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-lifecycle`

L'exemple de code suivant montre comment utiliser`get-bucket-lifecycle`.

AWS CLI

La commande suivante permet de récupérer la configuration du cycle de vie d'un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-lifecycle --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLifecycleà la section Référence des AWS CLI commandes.

`get-bucket-location`

L'exemple de code suivant montre comment utiliser`get-bucket-location`.

AWS CLI

La commande suivante récupère la contrainte d'emplacement pour un compartiment nomméamzn-s3-demo-bucket, s'il en existe une :

```
aws s3api get-bucket-location --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "LocationConstraint": "us-west-2"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLocationà la section Référence des AWS CLI commandes.

`get-bucket-logging`

L'exemple de code suivant montre comment utiliser`get-bucket-logging`.

AWS CLI

Pour récupérer l'état de journalisation d'un bucket

L'`get-bucket-logging`exemple suivant permet de récupérer l'état de journalisation pour le compartiment spécifié.

```
aws s3api get-bucket-logging \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "LoggingEnabled": {
```

```
        "TargetPrefix": "",
        "TargetBucket": "amzn-s3-demo-bucket-logs"
          }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLoggingà la section Référence des AWS CLI commandes.

`get-bucket-metrics-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-metrics-configuration`.

AWS CLI

Pour récupérer la configuration des métriques d'un bucket avec un ID spécifique

L'`get-bucket-metrics-configuration`exemple suivant affiche la configuration des métriques pour le bucket et l'ID spécifiés.

```
aws s3api get-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

Sortie :

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-notification-configuration`

L'exemple de code suivant montre comment utiliser`get-bucket-notification-configuration`.

AWS CLI

La commande suivante permet de récupérer la configuration des notifications pour un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "TopicConfigurations": [
        {
            "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketNotificationConfigurationà la section Référence des AWS CLI commandes.

`get-bucket-notification`

L'exemple de code suivant montre comment utiliser`get-bucket-notification`.

AWS CLI

La commande suivante permet de récupérer la configuration des notifications pour un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-notification --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
```

```
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketNotificationà la section Référence des AWS CLI commandes.

`get-bucket-ownership-controls`

L'exemple de code suivant montre comment utiliser`get-bucket-ownership-controls`.

AWS CLI

Pour récupérer les paramètres de propriété d'un bucket

L'`get-bucket-ownership-controls`exemple suivant récupère les paramètres de propriété d'un bucket.

```
aws s3api get-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "OwnershipControls": {
        "Rules": [
            {
                "ObjectOwnership": "BucketOwnerEnforced"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Affichage du paramètre de propriété de l'objet pour un compartiment S3 dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous GetBucketOwnershipControlsà la section Référence des AWS CLI commandes.

`get-bucket-policy-status`

L'exemple de code suivant montre comment utiliser`get-bucket-policy-status`.

AWS CLI

Pour récupérer le statut de la politique d'un compartiment indiquant s'il est public

L'`get-bucket-policy-status`exemple suivant permet de récupérer le statut de la politique pour le compartiment`amzn-s3-demo-bucket`.

```
aws s3api get-bucket-policy-status \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyStatusà la section Référence des AWS CLI commandes.

`get-bucket-policy`

L'exemple de code suivant montre comment utiliser`get-bucket-policy`.

AWS CLI

La commande suivante permet de récupérer la politique de compartiment pour un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
```

```
    "Policy": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect
\":\"Allow\",\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":
\"arn:aws:s3:::amzn-s3-demo-bucket/*\"},{\"Sid\":\"\",\"Effect\":\"Deny\",
\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::amzn-s3-
demo-bucket/secret/*\"}]}"
}
```

Obtenir et mettre une politique de compartiment L'exemple suivant montre comment télécharger une politique de compartiment Amazon S3, apporter des modifications au fichier, puis utiliser `put-bucket-policy` pour appliquer la politique de compartiment modifiée. Pour télécharger la politique du bucket dans un fichier, vous pouvez exécuter :

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket --query Policy --output
 text > policy.json
```

Vous pouvez ensuite modifier le `policy.json` fichier selon vos besoins. Enfin, vous pouvez réappliquer cette politique modifiée au compartiment S3 en exécutant :

`policy.json`fichier selon les besoins. Enfin, vous pouvez réappliquer cette politique modifiée au compartiment S3 en exécutant :

fichier selon les besoins. Enfin, vous pouvez réappliquer cette politique modifiée au compartiment S3 en exécutant :

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS CLI commandes.

`get-bucket-replication`

L'exemple de code suivant montre comment utiliser`get-bucket-replication`.

AWS CLI

La commande suivante permet de récupérer la configuration de réplication pour un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-replication --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "ReplicationConfiguration": {
        "Rules": [
            {
                "Status": "Enabled",
                "Prefix": "",
                "Destination": {
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket-backup",
                    "StorageClass": "STANDARD"
                },
                "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
            }
        ],
        "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketReplicationà la section Référence des AWS CLI commandes.

`get-bucket-request-payment`

L'exemple de code suivant montre comment utiliser`get-bucket-request-payment`.

AWS CLI

Pour récupérer la configuration de paiement des demandes pour un bucket

L'`get-bucket-request-payment`exemple suivant extrait la configuration payante du demandeur pour le compartiment spécifié.

```
aws s3api get-bucket-request-payment \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Payer": "BucketOwner"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketRequestPaymentà la section Référence des AWS CLI commandes.

`get-bucket-tagging`

L'exemple de code suivant montre comment utiliser`get-bucket-tagging`.

AWS CLI

La commande suivante permet de récupérer la configuration de balisage pour un compartiment nommé : `amzn-s3-demo-bucket`

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "TagSet": [
        {
            "Value": "marketing",
            "Key": "organization"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketTaggingà la section Référence des AWS CLI commandes.

`get-bucket-versioning`

L'exemple de code suivant montre comment utiliser`get-bucket-versioning`.

AWS CLI

La commande suivante permet de récupérer la configuration de version pour un compartiment nommé : `amzn-s3-demo-bucket`

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Status": "Enabled"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketVersioningà la section Référence des AWS CLI commandes.

`get-bucket-website`

L'exemple de code suivant montre comment utiliser`get-bucket-website`.

AWS CLI

La commande suivante permet de récupérer la configuration statique du site Web pour un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-bucket-website --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketWebsiteà la section Référence des AWS CLI commandes.

`get-object-acl`

L'exemple de code suivant montre comment utiliser`get-object-acl`.

AWS CLI

La commande suivante permet de récupérer la liste de contrôle d'accès pour un objet dans un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket --key index.html
```

Sortie :

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectAclà la section Référence des AWS CLI commandes.

`get-object-attributes`

L'exemple de code suivant montre comment utiliser`get-object-attributes`.

AWS CLI

Pour récupérer les métadonnées d'un objet sans renvoyer l'objet lui-même

L'`get-object-attributes`exemple suivant extrait les métadonnées de l'objet`doc1.rtf`.

```
aws s3api get-object-attributes \
    --bucket amzn-s3-demo-bucket \
```

```
    --key doc1.rtf \
    --object-attributes "StorageClass" "ETag" "ObjectSize"
```

Sortie :

```
{
    "LastModified": "2022-03-15T19:37:31+00:00",
    "VersionId": "IuCPjXTDzHNfldAuitVBIKJpF2p1fg4P",
    "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
    "StorageClass": "STANDARD",
    "ObjectSize": 405
}
```

Pour plus d'informations, consultez GetObjectAttributesle manuel de référence des API Amazon S3.

- Pour plus de détails sur l'API, reportez-vous GetObjectAttributesà la section Référence des AWS CLI commandes.

`get-object-legal-hold`

L'exemple de code suivant montre comment utiliser`get-object-legal-hold`.

AWS CLI

Récupère le statut de conservation légale d'un objet

L'`get-object-legal-hold`exemple suivant permet de récupérer le statut Legal Hold pour l'objet spécifié.

```
aws s3api get-object-legal-hold \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

Sortie :

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS CLI commandes.

`get-object-lock-configuration`

L'exemple de code suivant montre comment utiliser`get-object-lock-configuration`.

AWS CLI

Pour récupérer une configuration de verrouillage d'objet pour un bucket

L'`get-object-lock-configuration`exemple suivant récupère la configuration du verrouillage d'objet pour le compartiment spécifié.

```
aws s3api get-object-lock-configuration \
    --bucket amzn-s3-demo-bucket-with-object-lock
```

Sortie :

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS CLI commandes.

`get-object-retention`

L'exemple de code suivant montre comment utiliser`get-object-retention`.

AWS CLI

Pour récupérer la configuration de rétention d'un objet

L'`get-object-retention`exemple suivant récupère la configuration de rétention d'objets pour l'objet spécifié.

```
aws s3api get-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

Sortie :

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS CLI commandes.

`get-object-tagging`

L'exemple de code suivant montre comment utiliser`get-object-tagging`.

AWS CLI

Pour récupérer les tags attachés à un objet

L'`get-object-tagging`exemple suivant extrait les valeurs de la clé spécifiée à partir de l'objet spécifié.

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf
```

Sortie :

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
```

```
        }
    ]
}
```

L'`get-object-tagging`exemple suivant essaie de récupérer les ensembles de balises de l'objet`doc2.rtf`, qui ne possède aucune balise.

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc2.rtf
```

Sortie :

```
{
    "TagSet": []
}
```

L'`get-object-tagging`exemple suivant récupère les ensembles de balises de l'objet`doc3.rtf`, qui possède plusieurs balises.

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc3.rtf
```

Sortie :

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        },
        {
            "Value": "finance",
            "Key": "department"
        },
        {
            "Value": "payroll",
            "Key": "team"
        }
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectTaggingà la section Référence des AWS CLI commandes.

`get-object-torrent`

L'exemple de code suivant montre comment utiliser`get-object-torrent`.

AWS CLI

La commande suivante crée un torrent pour un objet dans un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api get-object-torrent --bucket amzn-s3-demo-bucket --key large-video-file.mp4 large-video-file.torrent
```

Le fichier torrent est enregistré localement dans le dossier en cours. Notez que le nom du fichier de sortie (`large-video-file.torrent`) est spécifié sans nom d'option et doit être le dernier argument de la commande.

- Pour plus de détails sur l'API, reportez-vous GetObjectTorrentà la section Référence des AWS CLI commandes.

`get-object`

L'exemple de code suivant montre comment utiliser`get-object`.

AWS CLI

L'exemple suivant utilise la `get-object` commande pour télécharger un objet depuis Amazon S3 :

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2 my_images.tar.bz2
```

Notez que le paramètre outfile est spécifié sans nom d'option tel que « --outfile ». Le nom du fichier de sortie doit être le dernier paramètre de la commande.

L'exemple ci-dessous illustre l'utilisation de `--range` pour télécharger une plage d'octets spécifique à partir d'un objet. Notez que les plages d'octets doivent être préfixées par « bytes= » :

```
aws s3api get-object --bucket text-content --key dir/my_data --
range bytes=8888-9999 my_data_range
```

Pour plus d'informations sur la récupération d'objets, consultez Getting Objects dans le manuel Amazon S3 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

`get-public-access-block`

L'exemple de code suivant montre comment utiliser`get-public-access-block`.

AWS CLI

Pour définir ou modifier la configuration de blocage de l'accès public pour un bucket

L'`get-public-access-block`exemple suivant montre la configuration de blocage de l'accès public pour le compartiment spécifié.

```
aws s3api get-public-access-block \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPublicAccessBlockà la section Référence des AWS CLI commandes.

`head-bucket`

L'exemple de code suivant montre comment utiliser`head-bucket`.

AWS CLI

La commande suivante vérifie l'accès à un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api head-bucket --bucket amzn-s3-demo-bucket
```

Si le bucket existe et que vous y avez accès, aucune sortie n'est renvoyée. Dans le cas contraire, un message d'erreur s'affichera. Par exemple :

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Pour plus de détails sur l'API, reportez-vous HeadBucketà la section Référence des AWS CLI commandes.

`head-object`

L'exemple de code suivant montre comment utiliser`head-object`.

AWS CLI

La commande suivante permet de récupérer les métadonnées d'un objet dans un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html
```

Sortie :

```
{
    "AcceptRanges": "bytes",
    "ContentType": "text/html",
    "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
    "ContentLength": 77,
    "VersionId": "null",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "Metadata": {}
}
```

- Pour plus de détails sur l'API, reportez-vous HeadObjectà la section Référence des AWS CLI commandes.

`list-bucket-analytics-configurations`

L'exemple de code suivant montre comment utiliser`list-bucket-analytics-configurations`.

AWS CLI

Pour récupérer la liste des configurations d'analyse pour un bucket

Ce qui suit `list-bucket-analytics-configurations` permet de récupérer la liste des configurations d'analyse pour le compartiment spécifié.

```
aws s3api list-bucket-analytics-configurations \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketAnalyticsConfigurationsà la section Référence des AWS CLI commandes.

`list-bucket-intelligent-tiering-configurations`

L'exemple de code suivant montre comment utiliser`list-bucket-intelligent-tiering-configurations`.

AWS CLI

Pour récupérer toutes les configurations S3 Intelligent-Tiering sur un bucket

L'`list-bucket-intelligent-tiering-configurations`exemple suivant récupère l'ensemble de la configuration S3 Intelligent-Tiering d'un bucket.

```
aws s3api list-bucket-intelligent-tiering-configurations \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "IsTruncated": false,
    "IntelligentTieringConfigurationList": [
        {
            "Id": "ExampleConfig",
            "Filter": {
                "Prefix": "images"
            },
            "Status": "Enabled",
            "Tierings": [
                {
                    "Days": 90,
                    "AccessTier": "ARCHIVE_ACCESS"
                },
                {
                    "Days": 180,
                    "AccessTier": "DEEP_ARCHIVE_ACCESS"
                }
            ]
        },
        {
            "Id": "ExampleConfig2",
            "Status": "Disabled",
            "Tierings": [
                {
                    "Days": 730,
                    "AccessTier": "ARCHIVE_ACCESS"
                }
            ]
        },
        {
            "Id": "ExampleConfig3",
            "Filter": {
                "Tag": {
                    "Key": "documents",
                    "Value": "taxes"
                }
            },
```

```
            "Status": "Enabled",
            "Tierings": [
                {
                    "Days": 90,
                    "AccessTier": "ARCHIVE_ACCESS"
                },
                {
                    "Days": 365,
                    "AccessTier": "DEEP_ARCHIVE_ACCESS"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation de S3 Intelligent-Tiering dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous ListBucketIntelligentTieringConfigurationsà la section Référence des AWS CLI commandes.

`list-bucket-inventory-configurations`

L'exemple de code suivant montre comment utiliser`list-bucket-inventory-configurations`.

AWS CLI

Pour récupérer la liste des configurations d'inventaire pour un bucket

L'`list-bucket-inventory-configurations`exemple suivant répertorie les configurations d'inventaire pour le compartiment spécifié.

```
aws s3api list-bucket-inventory-configurations \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "InventoryConfigurationList": [
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
```

```
                    "Format": "ORC",
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "1",
            "Schedule": {
                "Frequency": "Weekly"
            }
        },
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "CSV",
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "2",
            "Schedule": {
                "Frequency": "Daily"
            }
        }
    ],
    "IsTruncated": false
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketInventoryConfigurationsà la section Référence des AWS CLI commandes.

`list-bucket-metrics-configurations`

L'exemple de code suivant montre comment utiliser`list-bucket-metrics-configurations`.

AWS CLI

Pour récupérer la liste des configurations de métriques pour un bucket

L'`list-bucket-metrics-configurations`exemple suivant extrait une liste de configurations de métriques pour le compartiment spécifié.

```
aws s3api list-bucket-metrics-configurations \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketMetricsConfigurationsà la section Référence des AWS CLI commandes.

`list-buckets`

L'exemple de code suivant montre comment utiliser`list-buckets`.

AWS CLI

La commande suivante utilise la `list-buckets` commande pour afficher les noms de tous vos compartiments Amazon S3 (dans toutes les régions) :

```
aws s3api list-buckets --query "Buckets[].Name"
```

L'option de requête filtre la sortie `list-buckets` jusqu'aux seuls noms de compartiments.

Pour plus d'informations sur les compartiments, consultez la section Utilisation des compartiments Amazon S3 dans le manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS CLI commandes.

`list-multipart-uploads`

L'exemple de code suivant montre comment utiliser`list-multipart-uploads`.

AWS CLI

La commande suivante répertorie tous les téléchargements partitionnés actifs pour un bucket nommé : `amzn-s3-demo-bucket`

```
aws s3api list-multipart-uploads --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Uploads": [
        {
            "Initiator": {
                "DisplayName": "username",
                "ID": "arn:aws:iam::0123456789012:user/username"
            },
            "Initiated": "2015-06-02T18:01:30.000Z",
            "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
            "StorageClass": "STANDARD",
            "Key": "multipart/01",
            "Owner": {
                "DisplayName": "aws-account-name",
                "ID":
 "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
            }
        }
    ],
    "CommonPrefixes": []
}
```

Les téléchargements partitionnés en cours entraînent des coûts de stockage dans Amazon S3. Terminez ou annulez un téléchargement en plusieurs parties actif pour en supprimer certaines parties de votre compte.

- Pour plus de détails sur l'API, reportez-vous ListMultipartUploadsà la section Référence des AWS CLI commandes.

`list-object-versions`

L'exemple de code suivant montre comment utiliser`list-object-versions`.

AWS CLI

La commande suivante permet de récupérer les informations de version d'un objet dans un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket --prefix index.html
```

Sortie :

```
{
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": true,
            "VersionId": "B2VsEK5saUNNHKcOAJj7hIE86RozToyq",
            "Key": "index.html",
            "LastModified": "2015-11-10T00:57:03.000Z"
        },
        {
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
            "Key": "index.html",
            "LastModified": "2015-11-09T23:32:20.000Z"
        }
    ],
    "Versions": [
```

```
        {
            "LastModified": "2015-11-10T00:20:11.000Z",
            "VersionId": "Rb_l2T8UHDkFEwCgJjhlgPOZC0qJ.vpD",
            "ETag": "\"0622528de826c0df5db1258a23b80be5\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 38
        },
        {
            "LastModified": "2015-11-09T23:26:41.000Z",
            "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAFf",
            "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 38
        },
        {
            "LastModified": "2015-11-09T22:50:50.000Z",
            "VersionId": "null",
            "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 533823
        }
    ]
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListObjectVersionsà la section Référence des AWS CLI commandes.

`list-objects-v2`

L'exemple de code suivant montre comment utiliser`list-objects-v2`.

AWS CLI

Pour obtenir la liste des objets d'un bucket

L'`list-objects-v2`exemple suivant répertorie les objets contenus dans le compartiment spécifié.

```
aws s3api list-objects-v2 \
    --bucket amzn-s3-demo-bucket
```

Sortie :

```
{
    "Contents": [
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"621503c373607d548b37cff8778d992c\"",
            "StorageClass": "STANDARD",
            "Key": "doc1.rtf",
            "Size": 391
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",
            "StorageClass": "STANDARD",
            "Key": "doc2.rtf",
            "Size": 373
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"08210852f65a2e9cb999972539a64d68\"",
            "StorageClass": "STANDARD",
```

```
            "Key": "doc3.rtf",
            "Size": 399
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"d1852dd683f404306569471af106988e\"",
            "StorageClass": "STANDARD",
            "Key": "doc4.rtf",
            "Size": 6225
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans AWS CLI Command Reference.

`list-objects`

L'exemple de code suivant montre comment utiliser`list-objects`.

AWS CLI

L'exemple suivant utilise la commande `list-objects` pour afficher le nom de tous les objets du compartiment spécifié :

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

L'exemple utilise l'argument `--query` pour filtrer la sortie `list-objects` jusqu'à la valeur de clé et la taille de chaque objet.

Pour plus d'informations sur les objets, consultez Utilisation des objets Amazon S3 dans le Manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous ListObjectsà la section Référence des AWS CLI commandes.

`list-parts`

L'exemple de code suivant montre comment utiliser`list-parts`.

AWS CLI

La commande suivante répertorie toutes les parties qui ont été téléchargées pour un téléchargement en plusieurs parties avec la clé `multipart/01` dans le compartiment `amzn-s3-demo-bucket` :

```
aws s3api list-parts --bucket amzn-s3-demo-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

Sortie :

```
{
    "Owner": {
        "DisplayName": "aws-account-name",
        "ID": "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
    },
    "Initiator": {
        "DisplayName": "username",
        "ID": "arn:aws:iam::0123456789012:user/username"
    },
    "Parts": [
        {
            "LastModified": "2015-06-02T18:07:35.000Z",
            "PartNumber": 1,
            "ETag": "\"e868e0f4719e394144ef36531ee6824c\"",
            "Size": 5242880
        },
        {
            "LastModified": "2015-06-02T18:07:42.000Z",
            "PartNumber": 2,
            "ETag": "\"6bb2b12753d66fe86da4998aa33fffb0\"",
            "Size": 5242880
        },
        {
            "LastModified": "2015-06-02T18:07:47.000Z",
            "PartNumber": 3,
            "ETag": "\"d0a0112e841abec9c9ec83406f0159c8\"",
            "Size": 5242880
        }
    ],
    "StorageClass": "STANDARD"
}
```

- Pour plus de détails sur l'API, reportez-vous ListPartsà la section Référence des AWS CLI commandes.

`ls`

L'exemple de code suivant montre comment utiliser`ls`.

AWS CLI

Exemple 1 : liste de tous les compartiments appartenant à l'utilisateur

La `ls` commande suivante répertorie tous les compartiments appartenant à l'utilisateur. Dans cet exemple, l'utilisateur possède les compartiments `amzn-s3-demo-bucket` et`amzn-s3-demo-bucket2`. L'horodatage est la date à laquelle le bucket a été créé, indiquée dans le fuseau horaire de votre machine. Cette date peut changer lorsque vous apportez des modifications à votre compartiment, par exemple en modifiant sa politique en matière de compartiment. Notez que s'il `s3://` est utilisé pour l'argument path<S3Uri>, il listera également tous les compartiments.

```
aws s3 ls
```

Sortie :

```
2013-07-11 17:08:50 amzn-s3-demo-bucket
2013-07-24 14:55:44 amzn-s3-demo-bucket2
```

Exemple 2 : liste de tous les préfixes et objets d'un compartiment

La `ls` commande suivante répertorie les objets et les préfixes courants sous un compartiment et un préfixe spécifiés. Dans cet exemple, l'utilisateur est propriétaire du compartiment `amzn-s3-demo-bucket` contenant les objets `test.txt` et`somePrefix/test.txt`. Les `LastWriteTime` et `Length` sont arbitraires. Notez que puisque la `ls` commande n'a aucune interaction avec le système de fichiers local, le schéma d'`s3://`URI n'est pas nécessaire pour résoudre l'ambiguïté et peut être omis.

```
aws s3 ls s3://amzn-s3-demo-bucket
```

Sortie :

```
                           PRE somePrefix/
```

```
2013-07-25 17:06:27         88 test.txt
```

Exemple 3 : liste de tous les préfixes et objets d'un compartiment et d'un préfixe spécifiques

La `ls` commande suivante répertorie les objets et les préfixes courants sous un compartiment et un préfixe spécifiés. Cependant, il n'existe aucun objet ni préfixe commun sous le compartiment et le préfixe spécifiés.

```
aws s3 ls s3://amzn-s3-demo-bucket/noExistPrefix
```

Sortie :

```
None
```

Exemple 4 : liste récursive de tous les préfixes et objets d'un bucket

La `ls` commande suivante listera de manière récursive les objets d'un compartiment. Plutôt que de s'afficher `PRE dirname/` dans la sortie, tout le contenu d'un bucket sera répertorié dans l'ordre.

```
aws s3 ls s3://amzn-s3-demo-bucket \
    --recursive
```

Sortie :

```
2013-09-02 21:37:53         10 a.txt
2013-09-02 21:37:53    2863288 foo.zip
2013-09-02 21:32:57         23 foo/bar/.baz/a
2013-09-02 21:32:58         41 foo/bar/.baz/b
2013-09-02 21:32:57        281 foo/bar/.baz/c
2013-09-02 21:32:57         73 foo/bar/.baz/d
2013-09-02 21:32:57        452 foo/bar/.baz/e
2013-09-02 21:32:57        896 foo/bar/.baz/hooks/bar
2013-09-02 21:32:57        189 foo/bar/.baz/hooks/foo
2013-09-02 21:32:57        398 z.txt
```

Exemple 5 : Récapitulatif de tous les préfixes et objets d'un bucket

La `ls` commande suivante illustre la même commande à l'aide des options --human-readable et --summarize. --human-readable affiche la taille du fichier en. Bytes/MiB/KiB/GiB/TiB/PiB/EiB --summarise affiche le nombre total d'objets et leur taille totale à la fin de la liste des résultats :

```
aws s3 ls s3://amzn-s3-demo-bucket \
    --recursive \
    --human-readable \
    --summarize
```

Sortie :

```
2013-09-02 21:37:53   10 Bytes a.txt
2013-09-02 21:37:53  2.9 MiB foo.zip
2013-09-02 21:32:57   23 Bytes foo/bar/.baz/a
2013-09-02 21:32:58   41 Bytes foo/bar/.baz/b
2013-09-02 21:32:57  281 Bytes foo/bar/.baz/c
2013-09-02 21:32:57   73 Bytes foo/bar/.baz/d
2013-09-02 21:32:57  452 Bytes foo/bar/.baz/e
2013-09-02 21:32:57  896 Bytes foo/bar/.baz/hooks/bar
2013-09-02 21:32:57  189 Bytes foo/bar/.baz/hooks/foo
2013-09-02 21:32:57  398 Bytes z.txt

Total Objects: 10
   Total Size: 2.9 MiB
```

Exemple 6 : mise en vente depuis un point d'accès S3

La `ls` commande suivante répertorie les objets depuis le point d'accès (`myaccesspoint`) :

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Sortie :

```
                           PRE somePrefix/
2013-07-25 17:06:27         88 test.txt
```

- Pour plus de détails sur l'API, voir Ls dans AWS CLI Command Reference.

mb

L'exemple de code suivant montre comment utilisermb.

AWS CLI

Exemple 1 : créer un bucket

La mb commande suivante crée un bucket. Dans cet exemple, l'utilisateur crée le bucketamzn-s3-demo-bucket. Le bucket est créé dans la région spécifiée dans le fichier de configuration de l'utilisateur :

```
aws s3 mb s3://amzn-s3-demo-bucket
```

Sortie :

```
make_bucket: s3://amzn-s3-demo-bucket
```

Exemple 2 : créer un bucket dans la région spécifiée

La mb commande suivante crée un compartiment dans une région spécifiée par le --region paramètre. Dans cet exemple, l'utilisateur crée le bucket amzn-s3-demo-bucket dans la région us-west-1 :

```
aws s3 mb s3://amzn-s3-demo-bucket \
    --region us-west-1
```

Sortie :

```
make_bucket: s3://amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, voir Mb dans AWS CLI Command Reference.

mv

L'exemple de code suivant montre comment utilisermv.

AWS CLI

Exemple 1 : déplacer un fichier local vers le compartiment spécifié

La mv commande suivante déplace un seul fichier vers un compartiment et une clé spécifiés.

```
aws s3 mv test.txt s3://amzn-s3-demo-bucket/test2.txt
```

Sortie :

```
move: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 2 : déplacer un objet vers le compartiment et la clé spécifiés

La `mv` commande suivante déplace un seul objet s3 vers un compartiment et une clé spécifiés.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt
```

Sortie :

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 3 : déplacer un objet S3 vers le répertoire local

La `mv` commande suivante déplace localement un seul objet vers un fichier spécifié.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt test2.txt
```

Sortie :

```
move: s3://amzn-s3-demo-bucket/test.txt to test2.txt
```

Exemple 4 : déplacer un objet avec son nom d'origine vers le compartiment spécifié

La `mv` commande suivante déplace un seul objet vers un compartiment spécifié tout en conservant son nom d'origine :

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket2/
```

Sortie :

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

Exemple 5 : déplacer tous les objets et préfixes d'un bucket vers le répertoire local

Lorsqu'elle est passée avec le paramètre`--recursive`, la `mv` commande suivante déplace de manière récursive tous les objets sous un préfixe et un compartiment spécifiés vers un répertoire spécifié. Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et`test2.txt`.

```
aws s3 mv s3://amzn-s3-demo-bucket . \
    --recursive
```

Sortie :

```
move: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
move: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

Exemple 6 : déplacer tous les objets et préfixes d'un bucket vers le répertoire local, à l'exception des fichiers ``.jpg``

Lorsqu'elle est passée avec le paramètre`--recursive`, la `mv` commande suivante déplace de manière récursive tous les fichiers d'un répertoire spécifié vers un compartiment et un préfixe spécifiés, tout en excluant certains fichiers à l'aide d'un `--exclude` paramètre. Dans cet exemple, le répertoire `myDir` contient les fichiers `test1.txt` et`test2.jpg`.

```
aws s3 mv myDir s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

Sortie :

```
move: myDir/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

Exemple 7 : déplacer tous les objets et préfixes d'un bucket vers le répertoire local, à l'exception du préfixe spécifié

Lorsqu'elle est passée avec le paramètre`--recursive`, la `mv` commande suivante déplace de manière récursive tous les objets d'un compartiment spécifié vers un autre compartiment tout en excluant certains objets à l'aide d'un `--exclude` paramètre. Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et`another/test1.txt`.

```
aws s3 mv s3://amzn-s3-demo-bucket/ s3://amzn-s3-demo-bucket2/ \
    --recursive \
    --exclude "amzn-s3-demo-bucket/another/*"
```

Sortie :

```
move: s3://amzn-s3-demo-bucket/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

Exemple 8 : déplacer un objet vers le compartiment spécifié et définir l'ACL

La `mv` commande suivante déplace un seul objet vers un compartiment et une clé spécifiés tout en définissant l'ACL sur`public-read-write`.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --acl public-read-write
```

Sortie :

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 9 : déplacer un fichier local vers le compartiment spécifié et accorder des autorisations

La `mv` commande suivante illustre l'utilisation de l'`--grants`option permettant d'accorder un accès en lecture à tous les utilisateurs et un contrôle total à un utilisateur spécifique identifié par son adresse e-mail.

```
aws s3 mv file.txt s3://amzn-s3-demo-bucket/ \
    --grants read=uri=http://acs.amazonaws.com/groups/global/
AllUsers full=emailaddress=user@example.com
```

Sortie :

```
move: file.txt to s3://amzn-s3-demo-bucket/file.txt
```

Exemple 10 : déplacer un fichier vers un point d'accès S3

La `mv` commande suivante déplace un seul fichier nommé `mydoc.txt` vers le point d'accès nommé `myaccesspoint` sur la clé nomméemykey.

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

Sortie :

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey
```

- Pour plus de détails sur l'API, voir Mv dans AWS CLI Command Reference.

`presign`

L'exemple de code suivant montre comment utiliser`presign`.

AWS CLI

Exemple 1 : pour créer une URL pré-signée avec une durée de vie par défaut d'une heure qui renvoie à un objet dans un compartiment S3

La `presign` commande suivante génère une URL pré-signée pour un compartiment et une clé spécifiés, valide pendant une heure.

```
aws s3 presign s3://amzn-s3-demo-bucket/test2.txt
```

Sortie :

```
https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-
HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-
west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=3600&X-Amz-
SignedHeaders=host&X-Amz-
Signature=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

Exemple 2 : pour créer une URL pré-signée avec une durée de vie personnalisée qui renvoie à un objet dans un compartiment S3

La `presign` commande suivante génère une URL pré-signée pour un compartiment et une clé spécifiés, valide pendant une semaine.

```
aws s3 presign s3://amzn-s3-demo-bucket/test2.txt \
    --expires-in 604800
```

Sortie :

```
https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-
HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-
west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=604800&X-Amz-
SignedHeaders=host&X-Amz-
Signature=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

Pour plus d'informations, consultez Partager un objet avec d'autres dans le guide du développeur S3.

- Pour plus de détails sur l'API, consultez Presign in AWS CLI Command Reference.

`put-bucket-accelerate-configuration`

L'exemple de code suivant montre comment utiliser`put-bucket-accelerate-configuration`.

AWS CLI

Pour définir la configuration accélérée d'un bucket

L'`put-bucket-accelerate-configuration`exemple suivant active la configuration d'accélération pour le compartiment spécifié.

```
aws s3api put-bucket-accelerate-configuration \
    --bucket amzn-s3-demo-bucket \
    --accelerate-configuration Status=Enabled
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketAccelerateConfigurationà la section Référence des AWS CLI commandes.

`put-bucket-acl`

L'exemple de code suivant montre comment utiliser`put-bucket-acl`.

AWS CLI

Cet exemple accorde `full control` l'`read`autorisation à deux AWS utilisateurs (user1@example.com et user2@example.com) et à tout le monde :

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Voir http://docs.aws.amazon. com/AmazonS3/latest/API/RESTBucketPUTacl.html pour plus de détails sur la personnalisation ACLs (les commandes ACL s3api, par exemple`put-bucket-acl`, utilisent la même notation abrégée pour les arguments).

- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section Référence des AWS CLI commandes.

`put-bucket-analytics-configuration`

L'exemple de code suivant montre comment utiliser`put-bucket-analytics-configuration`.

AWS CLI

Pour définir une configuration d'analyse pour le bucket

L'`put-bucket-analytics-configuration`exemple suivant configure les analyses pour le compartiment spécifié.

```
aws s3api put-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket --id 1 \
    --analytics-configuration '{"Id": "1","StorageClassAnalysis": {}}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

`put-bucket-cors`

L'exemple de code suivant montre comment utiliser`put-bucket-cors`.

AWS CLI

L'exemple suivant active `PUT`POST, et les `DELETE` demandes provenant de www.example.com, et active les `GET` demandes provenant de n'importe quel domaine :

```
aws s3api put-bucket-cors --bucket amzn-s3-demo-bucket --cors-configuration file://
cors.json

cors.json:
{
  "CORSRules": [
    {
      "AllowedOrigins": ["http://www.example.com"],
      "AllowedHeaders": ["*"],
```

```
      "AllowedMethods": ["PUT", "POST", "DELETE"],
      "MaxAgeSeconds": 3000,
      "ExposeHeaders": ["x-amz-server-side-encryption"]
    },
    {
      "AllowedOrigins": ["*"],
      "AllowedHeaders": ["Authorization"],
      "AllowedMethods": ["GET"],
      "MaxAgeSeconds": 3000
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketCorsà la section Référence des AWS CLI commandes.

`put-bucket-encryption`

L'exemple de code suivant montre comment utiliser`put-bucket-encryption`.

AWS CLI

Pour configurer le chiffrement côté serveur pour un bucket

L'`put-bucket-encryption`exemple suivant définit le AES256 chiffrement comme valeur par défaut pour le compartiment spécifié.

```
aws s3api put-bucket-encryption \
    --bucket amzn-s3-demo-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketEncryptionà la section Référence des AWS CLI commandes.

`put-bucket-intelligent-tiering-configuration`

L'exemple de code suivant montre comment utiliser`put-bucket-intelligent-tiering-configuration`.

AWS CLI

Pour mettre à jour une configuration S3 Intelligent-Tiering sur un bucket

L'`put-bucket-intelligent-tiering-configuration`exemple suivant met à jour une configuration S3 Intelligent-Tiering, nommée ExampleConfig, sur un bucket. La configuration fera passer les objets non accessibles sous le préfixe images vers Archive Access après 90 jours et Deep Archive Access après 180 jours.

```
aws s3api put-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-bucket \
    --id "ExampleConfig" \
    --intelligent-tiering-configuration file://intelligent-tiering-
configuration.json
```

Contenu de `intelligent-tiering-configuration.json` :

```
{
    "Id": "ExampleConfig",
    "Status": "Enabled",
    "Filter": {
        "Prefix": "images"
        },
    "Tierings": [
        {
            "Days": 90,
            "AccessTier": "ARCHIVE_ACCESS"
        },
        {
            "Days": 180,
            "AccessTier": "DEEP_ARCHIVE_ACCESS"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition de la propriété d'un objet sur un compartiment existant dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous PutBucketIntelligentTieringConfigurationà la section Référence des AWS CLI commandes.

put-bucket-inventory-configuration

L'exemple de code suivant montre comment utiliserput-bucket-inventory-configuration.

AWS CLI

Exemple 1 : pour définir une configuration d'inventaire pour un bucket

L'put-bucket-inventory-configurationexemple suivant définit un rapport d'inventaire hebdomadaire au format ORC pour le compartiment. amzn-s3-demo-bucket

```
aws s3api put-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1 \
    --inventory-configuration '{"Destination": { "S3BucketDestination":
 { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
 "Format": "ORC" }}, "IsEnabled": true, "Id": "1", "IncludedObjectVersions":
 "Current", "Schedule": { "Frequency": "Weekly" }}'
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour définir une configuration d'inventaire pour un bucket

L'put-bucket-inventory-configurationexemple suivant définit un rapport d'inventaire quotidien au format CSV pour le compartiment. amzn-s3-demo-bucket

```
aws s3api put-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 2 \
    --inventory-configuration '{"Destination": { "S3BucketDestination":
 { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
 "Format": "CSV" }}, "IsEnabled": true, "Id": "2", "IncludedObjectVersions":
 "Current", "Schedule": { "Frequency": "Daily" }}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketInventoryConfigurationà la section Référence des AWS CLI commandes.

put-bucket-lifecycle-configuration

L'exemple de code suivant montre comment utiliserput-bucket-lifecycle-configuration.

AWS CLI

La commande suivante applique une configuration de cycle de vie à un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api put-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
lifecycle-configuration  file://lifecycle.json
```

Le fichier `lifecycle.json` est un document JSON situé dans le dossier actuel qui définit deux règles :

```
{
    "Rules": [
        {
            "ID": "Move rotated logs to Glacier",
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2015-11-10T00:00:00.000Z",
                    "StorageClass": "GLACIER"
                }
            ]
        },
        {
            "Status": "Enabled",
            "Prefix": "",
            "NoncurrentVersionTransitions": [
                {
                    "NoncurrentDays": 2,
                    "StorageClass": "GLACIER"
                }
            ],
            "ID": "Move old versions to Glacier"
        }
    ]
}
```

La première règle déplace les fichiers avec le préfixe `rotated` vers Glacier à la date spécifiée. La deuxième règle déplace les anciennes versions d'objets vers Glacier lorsqu'elles ne sont plus actuelles. Pour plus d'informations sur les formats d'horodatage acceptables, consultez la section Spécification des valeurs de paramètres dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous PutBucketLifecycleConfigurationà la section Référence des AWS CLI commandes.

`put-bucket-lifecycle`

L'exemple de code suivant montre comment utiliser`put-bucket-lifecycle`.

AWS CLI

La commande suivante applique une configuration de cycle de vie au compartiment `amzn-s3-demo-bucket` :

```
aws s3api put-bucket-lifecycle --bucket amzn-s3-demo-bucket --lifecycle-
configuration file://lifecycle.json
```

Le fichier `lifecycle.json` est un document JSON situé dans le dossier actuel qui définit deux règles :

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
  ]
}
```

La première règle déplace les fichiers vers Amazon Glacier au bout de soixante jours. La deuxième règle supprime les fichiers d'Amazon S3 à la date spécifiée. Pour plus d'informations sur les formats d'horodatage acceptables, consultez la section Spécification des valeurs de paramètres dans le guide de l'utilisateur de la AWS CLI.

Chaque règle de l'exemple ci-dessus spécifie une politique (`Transition`ou`Expiration`) et un préfixe de fichier (nom de dossier) auxquels elle s'applique. Vous pouvez également créer une règle qui s'applique à l'intégralité d'un compartiment en spécifiant un préfixe vide :

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (all objects in bucket)",
      "Prefix": "",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketLifecycleà la section Référence des AWS CLI commandes.

`put-bucket-logging`

L'exemple de code suivant montre comment utiliser`put-bucket-logging`.

AWS CLI

Exemple 1 : pour définir la journalisation des politiques relatives aux compartiments

L'`put-bucket-logging`exemple suivant définit la politique de journalisation pour amzn-s3-demo-bucket. Tout d'abord, accordez au service de journalisation l'autorisation principale dans votre politique de compartiment à l'aide de la `put-bucket-policy` commande.

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://policy.json
```

Contenu de `policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {"Service": "logging.s3.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/Logs/*",
            "Condition": {
                "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},
                "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}
            }
        }
    ]
}
```

Pour appliquer la politique de journalisation, utilisez`put-bucket-logging`.

```
aws s3api put-bucket-logging \
    --bucket amzn-s3-demo-bucket \
    --bucket-logging-status file://logging.json
```

Contenu de `logging.json` :

```
{
     "LoggingEnabled": {
         "TargetBucket": "amzn-s3-demo-bucket",
         "TargetPrefix": "Logs/"
     }
 }
```

La `put-bucket-policy` commande est requise pour accorder `s3:PutObject` des autorisations au principal du service de journalisation.

Pour plus d'informations, consultez la section Journalisation de l'accès au serveur Amazon S3 dans le guide de l'utilisateur Amazon S3.

Exemple 2 : pour définir une politique de compartiment pour la journalisation de l'accès à un seul utilisateur

L'`put-bucket-logging`exemple suivant définit la politique de journalisation pour amzn-s3-demo-bucket. L' AWS utilisateur bob@example.com aura un contrôle total sur les fichiers journaux, et personne d'autre n'y aura accès. Tout d'abord, accordez l'autorisation S3 avec`put-bucket-acl`.

```
aws s3api put-bucket-acl \
    --bucket amzn-s3-demo-bucket \
    --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \
    --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

Appliquez ensuite la politique de journalisation à l'aide de`put-bucket-logging`.

```
aws s3api put-bucket-logging \
    --bucket amzn-s3-demo-bucket \
    --bucket-logging-status file://logging.json
```

Contenu de `logging.json` :

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-bucket",
        "TargetPrefix": "amzn-s3-demo-bucket-logs/",
        "TargetGrants": [
            {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "bob@example.com"
                },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

la `put-bucket-acl` commande est requise pour accorder au système de livraison de journaux de S3 les autorisations nécessaires (autorisations d'écriture et de lecture).

Pour plus d'informations, consultez la section Journalisation de l'accès au serveur Amazon S3 dans le manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous PutBucketLoggingà la section Référence des AWS CLI commandes.

`put-bucket-metrics-configuration`

L'exemple de code suivant montre comment utiliser`put-bucket-metrics-configuration`.

AWS CLI

Pour définir une configuration de métriques pour un bucket

L'`put-bucket-metrics-configuration`exemple suivant définit une configuration métrique avec l'ID 123 pour le compartiment spécifié.

```
aws s3api put-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

`put-bucket-notification-configuration`

L'exemple de code suivant montre comment utiliser`put-bucket-notification-configuration`.

AWS CLI

Pour activer les notifications spécifiées dans un bucket

L'`put-bucket-notification-configuration`exemple suivant applique une configuration de notification à un compartiment nomméamzn-s3-demo-bucket. Le fichier `notification.json` est un document JSON situé dans le dossier actuel qui spécifie une rubrique SNS et un type d'événement à surveiller.

```
aws s3api put-bucket-notification-configuration \
    --bucket amzn-s3-demo-bucket \
    --notification-configuration file://notification.json
```

Contenu de `notification.json` :

```
{
    "TopicConfigurations": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

La rubrique SNS doit être associée à une politique IAM qui autorise Amazon S3 à y publier.

```
{
    "Version": "2008-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "arn:aws:sns:us-west-2:123456789012::s3-notification-topic",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
                }
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationConfigurationà la section Référence des AWS CLI commandes.

`put-bucket-notification`

L'exemple de code suivant montre comment utiliser`put-bucket-notification`.

AWS CLI

Applique une configuration de notification à un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api put-bucket-notification --bucket amzn-s3-demo-bucket --notification-
configuration file://notification.json
```

Le fichier `notification.json` est un document JSON situé dans le dossier actuel qui indique une rubrique SNS et un type d'événement à surveiller :

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

La rubrique SNS doit être associée à une politique IAM qui autorise Amazon S3 à y publier :

```
{
 "Version": "2008-10-17",
 "Id": "example-ID",
 "Statement": [
  {
   "Sid": "example-statement-ID",
   "Effect": "Allow",
   "Principal": {
     "Service": "s3.amazonaws.com"
   },
   "Action": [
    "SNS:Publish"
   ],
   "Resource": "arn:aws:sns:us-west-2:123456789012:amzn-s3-demo-bucket",
   "Condition": {
      "ArnLike": {
      "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
    }
   }
  }
 ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationà la section Référence des AWS CLI commandes.

`put-bucket-ownership-controls`

L'exemple de code suivant montre comment utiliser`put-bucket-ownership-controls`.

AWS CLI

Pour mettre à jour les paramètres de propriété d'un bucket

L'`put-bucket-ownership-controls`exemple suivant met à jour les paramètres de propriété d'un compartiment.

```
aws s3api put-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket \
    --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Définition de la propriété d'un objet sur un compartiment existant dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous PutBucketOwnershipControlsà la section Référence des AWS CLI commandes.

`put-bucket-policy`

L'exemple de code suivant montre comment utiliser`put-bucket-policy`.

AWS CLI

Cet exemple permet à tous les utilisateurs de récupérer n'importe quel objet dans amzn-s3-demo-bucket, à l'exception de ceux du. MySecretFolder Il accorde `put` également une `delete` autorisation à l'utilisateur root du AWS compte `1234-5678-9012` :

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json

policy.json:
{
   "Statement": [
```

```
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      },
      {
         "Effect": "Deny",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/MySecretFolder/*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
         },
         "Action": [
            "s3:DeleteObject",
            "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }

   ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketPolicyà la section Référence des AWS CLI commandes.

`put-bucket-replication`

L'exemple de code suivant montre comment utiliser`put-bucket-replication`.

AWS CLI

Pour configurer la réplication pour un compartiment S3

L'`put-bucket-replication`exemple suivant applique une configuration de réplication au compartiment S3 spécifié.

```
aws s3api put-bucket-replication \
```

```
    --bucket amzn-s3-demo-bucket1 \
    --replication-configuration file://replication.json
```

Contenu de `replication.json` :

```
{
    "Role": "arn:aws:iam::123456789012:role/s3-replication-role",
    "Rules": [
        {
            "Status": "Enabled",
            "Priority": 1,
            "DeleteMarkerReplication": { "Status": "Disabled" },
            "Filter" : { "Prefix": ""},
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket2"
            }
        }
    ]
}
```

La gestion des versions doit être activée dans le compartiment de destination. Le rôle spécifié doit être autorisé à écrire dans le compartiment de destination et avoir une relation de confiance qui permet à Amazon S3 d'assumer le rôle.

Exemple de politique d'autorisation des rôles :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ReplicateTags"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket2/*"
        }
    ]
}
```

Exemple de politique de relation de confiance :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Ceci est le titre du sujet dans le guide de l'utilisateur de la console Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous PutBucketReplicationà la section Référence des AWS CLI commandes.

put-bucket-request-payment

L'exemple de code suivant montre comment utiliserput-bucket-request-payment.

AWS CLI

Exemple 1 : Pour activer la configuration ``requester pay`` pour un bucket

L'put-bucket-request-paymentexemple suivant active requester pays le bucket spécifié.

```
aws s3api put-bucket-request-payment \
    --bucket amzn-s3-demo-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

Cette commande ne produit aucun résultat.

Exemple 2 : Pour désactiver la configuration ``requester pay`` pour un bucket

L'put-bucket-request-paymentexemple suivant désactive requester pays le bucket spécifié.

```
aws s3api put-bucket-request-payment \
    --bucket amzn-s3-demo-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketRequestPaymentà la section Référence des AWS CLI commandes.

put-bucket-tagging

L'exemple de code suivant montre comment utiliserput-bucket-tagging.

AWS CLI

La commande suivante applique une configuration de balisage à un compartiment nommé amzn-s3-demo-bucket :

```
aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging file://
tagging.json
```

Le fichier `tagging.json` est un document JSON dans le dossier actuel qui spécifie les balises :

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

Ou appliquez une configuration de balisage `amzn-s3-demo-bucket` directement depuis la ligne de commande :

```
aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- Pour plus de détails sur l'API, reportez-vous PutBucketTaggingà la section Référence des AWS CLI commandes.

`put-bucket-versioning`

L'exemple de code suivant montre comment utiliser`put-bucket-versioning`.

AWS CLI

La commande suivante active le versionnement sur un bucket nommé `amzn-s3-demo-bucket` :

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled
```

La commande suivante active le versionnement et utilise un code mfa

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- Pour plus de détails sur l'API, reportez-vous PutBucketVersioningà la section Référence des AWS CLI commandes.

`put-bucket-website`

L'exemple de code suivant montre comment utiliser`put-bucket-website`.

AWS CLI

Applique une configuration de site Web statique à un compartiment nommé `amzn-s3-demo-bucket` :

```
aws s3api put-bucket-website --bucket amzn-s3-demo-bucket --website-
configuration file://website.json
```

Le fichier `website.json` est un document JSON situé dans le dossier actuel qui indique les pages d'index et d'erreur du site Web :

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS CLI commandes.

`put-object-acl`

L'exemple de code suivant montre comment utiliser`put-object-acl`.

AWS CLI

La commande suivante accorde l'`read`autorisation `full control` à deux AWS utilisateurs (user1@example.com et user2@example.com) et à tout le monde :

```
aws s3api put-object-acl --bucket amzn-s3-demo-bucket --key file.txt --grant-full-
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Voir http://docs.aws.amazon. com/AmazonS3/latest/API/RESTBucketPUTacl.html pour plus de détails sur la personnalisation ACLs (les commandes ACL s3api, par exemple`put-object-acl`, utilisent la même notation abrégée pour les arguments).

- Pour plus de détails sur l'API, reportez-vous PutObjectAclà la section Référence des AWS CLI commandes.

`put-object-legal-hold`

L'exemple de code suivant montre comment utiliser`put-object-legal-hold`.

AWS CLI

Pour appliquer une suspension légale à un objet

L'`put-object-legal-hold`exemple suivant définit une conservation légale sur l'objet`doc1.rtf`.

```
aws s3api put-object-legal-hold \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --legal-hold Status=ON
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectLegalHoldà la section Référence des AWS CLI commandes.

`put-object-lock-configuration`

L'exemple de code suivant montre comment utiliser`put-object-lock-configuration`.

AWS CLI

Pour définir une configuration de verrouillage d'objets sur un bucket

L'`put-object-lock-configuration`exemple suivant définit un verrouillage d'objet de 50 jours sur le compartiment spécifié.

```
aws s3api put-object-lock-configuration \
    --bucket amzn-s3-demo-bucket-with-object-lock \
```

```
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
{ "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectLockConfigurationà la section Référence des AWS CLI commandes.

`put-object-retention`

L'exemple de code suivant montre comment utiliser`put-object-retention`.

AWS CLI

Pour définir une configuration de rétention d'objets pour un objet

L'`put-object-retention`exemple suivant définit une configuration de rétention d'objet pour l'objet spécifié jusqu'au 01/01/2025.

```
aws s3api put-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS CLI commandes.

`put-object-tagging`

L'exemple de code suivant montre comment utiliser`put-object-tagging`.

AWS CLI

Pour définir une étiquette sur un objet

L'`put-object-tagging`exemple suivant définit une balise avec la clé `designation` et la valeur `confidential` sur l'objet spécifié.

```
aws s3api put-object-tagging \
    --bucket amzn-s3-demo-bucket \
```

```
    --key doc1.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" }]}'
```

Cette commande ne produit aucun résultat.

L'`put-object-tagging`exemple suivant définit plusieurs ensembles de balises sur l'objet spécifié.

```
aws s3api put-object-tagging \
    --bucket amzn-s3-demo-bucket-example \
    --key doc3.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" },
 { "Key": "department", "Value": "finance" }, { "Key": "team", "Value":
 "payroll" } ]}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectTaggingà la section Référence des AWS CLI commandes.

`put-object`

L'exemple de code suivant montre comment utiliser`put-object`.

AWS CLI

Exemple 1 : télécharger un objet sur Amazon S3

L'exemple de `put-object` commande suivant télécharge un objet sur Amazon S3.

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key my-dir/MySampleImage.png \
    --body MySampleImage.png
```

Pour plus d'informations sur le téléchargement d'objets, voir Chargement d'objets < http://docs.aws.amazon. com/AmazonS3/latest/dev/UploadingObjects.html> dans le manuel du développeur Amazon S3.

Exemple 2 : télécharger un fichier vidéo sur Amazon S3

L'exemple de `put-object` commande suivant permet de télécharger un fichier vidéo.

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key my-dir/big-video-file.mp4 \
    --body /media/videos/f-sharp-3-data-services.mp4
```

Pour plus d'informations sur le téléchargement d'objets, voir Chargement d'objets < http://docs.aws.amazon. com/AmazonS3/latest/dev/UploadingObjects.html> dans le manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

`put-public-access-block`

L'exemple de code suivant montre comment utiliser`put-public-access-block`.

AWS CLI

Pour définir la configuration de blocage de l'accès public pour un bucket

L'`put-public-access-block`exemple suivant définit une configuration d'accès public par blocs restrictifs pour le compartiment spécifié.

```
aws s3api put-public-access-block \
    --bucket amzn-s3-demo-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutPublicAccessBlockà la section Référence des AWS CLI commandes.

`rb`

L'exemple de code suivant montre comment utiliser`rb`.

AWS CLI

Exemple 1 : supprimer un bucket

La `rb` commande suivante supprime un bucket. Dans cet exemple, le bucket de l'utilisateur est`amzn-s3-demo-bucket`. Notez que le compartiment doit être vide pour pouvoir être retiré :

```
aws s3 rb s3://amzn-s3-demo-bucket
```

Sortie :

```
remove_bucket: amzn-s3-demo-bucket
```

Exemple 2 : Forcer la suppression d'un bucket

La `rb` commande suivante utilise le `--force` paramètre pour supprimer d'abord tous les objets du compartiment, puis pour retirer le compartiment lui-même. Dans cet exemple, le bucket de l'utilisateur est `amzn-s3-demo-bucket` et les objets qu'il contient `amzn-s3-demo-bucket` sont `test1.txt` et `test2.txt` :

```
aws s3 rb s3://amzn-s3-demo-bucket \
    --force
```

Sortie :

```
delete: s3://amzn-s3-demo-bucket/test1.txt
delete: s3://amzn-s3-demo-bucket/test2.txt
remove_bucket: amzn-s3-demo-bucket
```

- Pour plus de détails sur l'API, voir Rb dans AWS CLI Command Reference.

`restore-object`

L'exemple de code suivant montre comment utiliser`restore-object`.

AWS CLI

Pour créer une demande de restauration pour un objet

L'`restore-object`exemple suivant restaure l'objet Amazon S3 Glacier spécifié pour le compartiment `my-glacier-bucket` pendant 10 jours.

```
aws s3api restore-object \
    --bucket my-glacier-bucket \
```

```
    --key doc1.rtf \
    --restore-request Days=10
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RestoreObjectà la section Référence des AWS CLI commandes.

`rm`

L'exemple de code suivant montre comment utiliser`rm`.

AWS CLI

Exemple 1 : Supprimer un objet S3

La `rm` commande suivante supprime un seul objet s3 :

```
aws s3 rm s3://amzn-s3-demo-bucket/test2.txt
```

Sortie :

```
delete: s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 2 : supprimer tout le contenu d'un bucket

La `rm` commande suivante supprime de manière récursive tous les objets d'un compartiment et d'un préfixe spécifiés lorsqu'elle est transmise avec le paramètre. `--recursive` Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et `test2.txt` :

```
aws s3 rm s3://amzn-s3-demo-bucket \
    --recursive
```

Sortie :

```
delete: s3://amzn-s3-demo-bucket/test1.txt
delete: s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 3 : Supprimer tout le contenu d'un bucket, à l'exception des fichiers ``.jpg``

La `rm` commande suivante supprime de manière récursive tous les objets d'un compartiment et d'un préfixe spécifiés lorsqu'elle est transmise avec le paramètre, `--recursive` tout en excluant certains objets à l'aide d'un paramètre. `--exclude` Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et `test2.jpg` :

```
aws s3 rm s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

Sortie :

```
delete: s3://amzn-s3-demo-bucket/test1.txt
```

Exemple 4 : Supprimer tout le contenu d'un compartiment, à l'exception des objets sous le préfixe spécifié

La `rm` commande suivante supprime de manière récursive tous les objets d'un compartiment et d'un préfixe spécifiés lorsqu'elle est transmise avec le paramètre, `--recursive` tout en excluant tous les objets sous un préfixe particulier à l'aide d'un paramètre. `--exclude` Dans cet exemple, le compartiment `amzn-s3-demo-bucket` contient les objets `test1.txt` et `another/test.txt` :

```
aws s3 rm s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "another/*"
```

Sortie :

```
delete: s3://amzn-s3-demo-bucket/test1.txt
```

Exemple 5 : Supprimer un objet d'un point d'accès S3

La `rm` commande suivante supprime un seul objet (`mykey`) du point d'accès (`myaccesspoint`). : : La `rm` commande suivante supprime un seul objet (`mykey`) du point d'accès (`myaccesspoint`).

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

Sortie :

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

- Pour plus de détails sur l'API, voir Rm dans AWS CLI Command Reference.

`select-object-content`

L'exemple de code suivant montre comment utiliser`select-object-content`.

AWS CLI

Pour filtrer le contenu d'un objet Amazon S3 en fonction d'une instruction SQL

L'`select-object-content`exemple suivant filtre l'objet `my-data-file.csv` avec l'instruction SQL spécifiée et envoie la sortie dans un fichier.

```
aws s3api select-object-content \
    --bucket amzn-s3-demo-bucket \
    --key my-data-file.csv \
    --expression "select * from s3object limit 100" \
    --expression-type 'SQL' \
    --input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \
    --output-serialization '{"CSV": {}}' "output.csv"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SelectObjectContentà la section Référence des AWS CLI commandes.

`sync`

L'exemple de code suivant montre comment utiliser`sync`.

AWS CLI

Exemple 1 : synchronisation de tous les objets locaux avec le compartiment spécifié

La `sync` commande suivante synchronise les objets d'un répertoire local avec le préfixe et le compartiment spécifiés en chargeant les fichiers locaux vers S3. Un fichier local devra être chargé si sa taille est différente de celle de l'objet S3, si l'heure de dernière modification du fichier local est plus récente que l'heure de dernière modification de l'objet S3 ou si le fichier local n'existe pas

sous le compartiment et le préfixe spécifiés. Dans cet exemple, l'utilisateur synchronise le bucket avec `amzn-s3-demo-bucket` le répertoire courant local. Le répertoire courant local contient les fichiers `test.txt` et`test2.txt`. Le compartiment ne `amzn-s3-demo-bucket` contient aucun objet.

```
aws s3 sync . s3://amzn-s3-demo-bucket
```

Sortie :

```
upload: test.txt to s3://amzn-s3-demo-bucket/test.txt
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 2 : Synchroniser tous les objets S3 du compartiment S3 spécifié vers un autre compartiment

La `sync` commande suivante synchronise les objets sous un préfixe et un compartiment spécifiés avec les objets sous un autre préfixe et compartiment spécifiés en copiant des objets S3. Un objet S3 devra être copié si les tailles des deux objets S3 sont différentes, si l'heure de dernière modification de la source est plus récente que l'heure de dernière modification de la destination ou si l'objet S3 n'existe pas dans le compartiment et le préfixe de destination spécifiés.

Dans cet exemple, l'utilisateur synchronise le bucket avec `amzn-s3-demo-bucket` le bucket`amzn-s3-demo-bucket2`. Le compartiment `amzn-s3-demo-bucket` contient les objets `test.txt` et`test2.txt`. Le bucket ne `amzn-s3-demo-bucket2` contient aucun objet :

```
aws s3 sync s3://amzn-s3-demo-bucket s3://amzn-s3-demo-bucket2
```

Sortie :

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
copy: s3://amzn-s3-demo-bucket/test2.txt to s3://amzn-s3-demo-bucket2/test2.txt
```

Exemple 3 : Synchroniser tous les objets S3 du compartiment S3 spécifié vers le répertoire local

La `sync` commande suivante synchronise les fichiers du compartiment S3 spécifié vers le répertoire local en téléchargeant des objets S3. Un objet S3 devra être téléchargé si sa taille est différente de celle du fichier local, si l'heure de dernière modification de l'objet S3 est plus récente que l'heure de dernière modification du fichier local ou si l'objet S3 n'existe pas dans le

répertoire local. Notez que lorsque des objets sont téléchargés depuis S3, l'heure de dernière modification du fichier local est remplacée par l'heure de dernière modification de l'objet S3. Dans cet exemple, l'utilisateur synchronise le bucket avec `amzn-s3-demo-bucket` le répertoire local actuel. Le compartiment `amzn-s3-demo-bucket` contient les objets `test.txt` et`test2.txt`. Le répertoire local actuel ne contient aucun fichier :

```
aws s3 sync s3://amzn-s3-demo-bucket .
```

Sortie :

```
download: s3://amzn-s3-demo-bucket/test.txt to test.txt
download: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

Exemple 4 : Synchroniser tous les objets locaux avec le compartiment spécifié et supprimer tous les fichiers qui ne correspondent pas

La `sync` commande suivante synchronise les objets sous un préfixe et un compartiment spécifiés avec les fichiers d'un répertoire local en téléchargeant les fichiers locaux sur S3. En raison de `--delete` ce paramètre, tous les fichiers existant sous le préfixe et le compartiment spécifiés mais n'existant pas dans le répertoire local seront supprimés. Dans cet exemple, l'utilisateur synchronise le bucket avec `amzn-s3-demo-bucket` le répertoire courant local. Le répertoire courant local contient les fichiers `test.txt` et`test2.txt`. Le bucket `amzn-s3-demo-bucket` contient l'objet `test3.txt` :

```
aws s3 sync . s3://amzn-s3-demo-bucket \
    --delete
```

Sortie :

```
upload: test.txt to s3://amzn-s3-demo-bucket/test.txt
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
delete: s3://amzn-s3-demo-bucket/test3.txt
```

Exemple 5 : Synchroniser tous les objets locaux avec le compartiment spécifié, à l'exception des fichiers ``.jpg``

La `sync` commande suivante synchronise les objets sous un préfixe et un compartiment spécifiés avec les fichiers d'un répertoire local en téléchargeant les fichiers locaux sur S3. En raison de

`--exclude` ce paramètre, tous les fichiers correspondant au modèle existant à la fois dans S3 et localement seront exclus de la synchronisation. Dans cet exemple, l'utilisateur synchronise le bucket avec `amzn-s3-demo-bucket` le répertoire courant local. Le répertoire courant local contient les fichiers `test.jpg` et`test2.txt`. Le bucket `amzn-s3-demo-bucket` contient l'objet `test.jpg` d'une taille différente de celle du compartiment local `test.jpg` :

```
aws s3 sync . s3://amzn-s3-demo-bucket \
    --exclude "*.jpg"
```

Sortie :

```
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
```

Exemple 6 : Synchroniser tous les objets locaux avec le compartiment spécifié, à l'exception des fichiers de répertoire spécifiés

La `sync` commande suivante synchronise les fichiers d'un répertoire local avec les objets dotés d'un préfixe et d'un compartiment spécifiés en téléchargeant des objets S3. Cet exemple utilise l'indicateur de `--exclude` paramètre pour exclure un répertoire et un préfixe S3 spécifiés de la `sync` commande. Dans cet exemple, l'utilisateur synchronise le répertoire courant local avec le bucket`amzn-s3-demo-bucket`. Le répertoire courant local contient les fichiers `test.txt` et`another/test2.txt`. Le bucket `amzn-s3-demo-bucket` contient les objets `another/test5.txt` et `test1.txt` :

```
aws s3 sync s3://amzn-s3-demo-bucket/ . \
    --exclude "*another/*"
```

Sortie :

```
download: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
```

Exemple 7 : Synchroniser tous les objets entre des compartiments situés dans différentes régions

La `sync` commande suivante synchronise les fichiers entre deux compartiments situés dans des régions différentes :

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \
    --source-region us-west-2 \
```

```
    --region us-east-1
```

Sortie :

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

Exemple 8 : synchronisation avec un point d'accès S3

La `sync` commande suivante synchronise le répertoire actuel avec le point d'accès (`myaccesspoint`) :

```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Sortie :

```
upload: test.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test.txt
upload: test2.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test2.txt
```

- Pour plus de détails sur l'API, voir Sync in AWS CLI Command Reference.

`upload-part-copy`

L'exemple de code suivant montre comment utiliser`upload-part-copy`.

AWS CLI

Pour télécharger une partie d'un objet en copiant les données d'un objet existant en tant que source de données

L'`upload-part-copy`exemple suivant télécharge une partie en copiant les données d'un objet existant en tant que source de données.

```
aws s3api upload-part-copy \
    --bucket amzn-s3-demo-bucket \
    --key "Map_Data_June.mp4" \
    --copy-source "amzn-s3-demo-bucket/copy_of_Map_Data_June.mp4" \
    --part-number 1 \
```

```
    --upload-
id "bq0tdE1CDpWQYRPLHuNG50xAT6pA5D.m_RiBy0ggOH6b13pVRY7QjvLlf75iFdJqp_2wztk5hvpUM2SesXgrzbeh
```

Sortie :

```
{
    "CopyPartResult": {
        "LastModified": "2019-12-13T23:16:03.000Z",
        "ETag": "\"711470fc377698c393d94aed6305e245\""
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UploadPartCopyà la section Référence des AWS CLI commandes.

`upload-part`

L'exemple de code suivant montre comment utiliser`upload-part`.

AWS CLI

La commande suivante télécharge la première partie d'un téléchargement en plusieurs parties initié par la `create-multipart-upload` commande :

```
aws s3api upload-part --bucket amzn-s3-demo-bucket --key
 'multipart/01' --part-number 1 --body part01 --upload-id
  "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

L'`body`option prend le nom ou le chemin d'un fichier local à télécharger (n'utilisez pas le préfixe file ://). La taille minimale des pièces est de 5 Mo. L'identifiant de téléchargement est renvoyé par `create-multipart-upload` et peut également être récupéré avec`list-multipart-uploads`. Le bucket et la clé sont spécifiés lorsque vous créez le téléchargement partitionné.

Sortie :

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

Enregistrez la ETag valeur de chaque pièce pour plus tard. Ils sont nécessaires pour effectuer le téléchargement en plusieurs parties.

- Pour plus de détails sur l'API, reportez-vous UploadPartà la section Référence des AWS CLI commandes.

`website`

L'exemple de code suivant montre comment utiliser`website`.

AWS CLI

Configuration d'un compartiment S3 en tant que site Web statique

La commande suivante configure un bucket `amzn-s3-demo-bucket` nommé site Web statique. L'option du document d'index indique le fichier vers `amzn-s3-demo-bucket` lequel les visiteurs seront dirigés lorsqu'ils accèderont à l'URL du site Web. Dans ce cas, le bucket se trouve dans la région us-west-2, le site apparaîtra donc à l'adresse. `http://amzn-s3-demo-bucket.s3-website-us-west-2.amazonaws.com`

Tous les fichiers du bucket qui apparaissent sur le site statique doivent être configurés pour permettre aux visiteurs de les ouvrir. Les autorisations sur les fichiers sont configurées séparément de la configuration du site Web du bucket.

```
aws s3 website s3://amzn-s3-demo-bucket/ \
    --index-document index.html \
    --error-document error.html
```

Pour plus d'informations sur l'hébergement d'un site Web statique dans Amazon S3, consultez la section Hébergement d'un site Web statique dans le manuel Amazon Simple Storage Service Developer Guide.

- Pour plus de détails sur l'API, voir Site Web dans AWS CLI Command Reference.

Exemples d'Amazon S3 Control utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon S3 Control.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-access-point`

L'exemple de code suivant montre comment utiliser`create-access-point`.

AWS CLI

Pour créer un point d'accès

L'`create-access-point`exemple suivant crée un point d'accès nommé `finance-ap` d'après le bucket du compte `business-records` 123456789012. Avant d'exécuter cet exemple, remplacez le nom du point d'accès, le nom du compartiment et le numéro de compte par les valeurs appropriées à votre cas d'utilisation.

```
aws s3control create-access-point \
    --account-id 123456789012 \
    --bucket business-records \
    --name finance-ap
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création de points d'accès dans le manuel Amazon Simple Storage Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateAccessPointà la section Référence des AWS CLI commandes.

`create-job`

L'exemple de code suivant montre comment utiliser`create-job`.

AWS CLI

Pour créer une tâche d'opérations par lots Amazon S3

L'`create-job`exemple suivant crée une tâche d'opérations par lots Amazon S3 pour étiqueter les objets comme tels`confidential` in the bucket ``employee-records`.

```
aws s3control create-job \
    --account-id 123456789012 \
    --operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"confidential",
 "Value":"true"}] }}' \
    --report '{"Bucket":"arn:aws:s3:::employee-records-logs","Prefix":"batch-op-
create-job",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::employee-records-logs/inv-
report/7a6a9be4-072c-407e-85a2-
ec3e982f773e.csv","ETag":"69f52a4e9f797e987155d9c8f5880897"}}' \
    --priority 42 \
    --role-arn arn:aws:iam::123456789012:role/S3BatchJobRole
```

Sortie :

```
{
    "JobId": "93735294-df46-44d5-8638-6356f335324e"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateJobà la section Référence des AWS CLI commandes.

`delete-access-point-policy`

L'exemple de code suivant montre comment utiliser`delete-access-point-policy`.

AWS CLI

Pour supprimer une politique de point d'accès

L'`delete-access-point-policy`exemple suivant supprime la politique de point d'accès du point d'accès nommé `finance-ap` dans le compte 123456789012. Avant d'exécuter cet exemple, remplacez le nom du point d'accès et le numéro de compte par des valeurs adaptées à votre cas d'utilisation.

```
aws s3control delete-access-point-policy \
    --account-id 123456789012 \
```

```
    --name finance-ap
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de l'accès aux données avec les points d'accès Amazon S3 dans le guide du développeur Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous DeleteAccessPointPolicyà la section Référence des AWS CLI commandes.

`delete-access-point`

L'exemple de code suivant montre comment utiliser`delete-access-point`.

AWS CLI

Pour supprimer un point d'accès

L'`delete-access-point`exemple suivant supprime un point d'accès nommé `finance-ap` dans le compte 123456789012. Avant d'exécuter cet exemple, remplacez le nom du point d'accès et le numéro de compte par des valeurs adaptées à votre cas d'utilisation.

```
aws s3control delete-access-point \
    --account-id 123456789012 \
    --name finance-ap
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de l'accès aux données avec les points d'accès Amazon S3 dans le guide du développeur Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous DeleteAccessPointà la section Référence des AWS CLI commandes.

`delete-public-access-block`

L'exemple de code suivant montre comment utiliser`delete-public-access-block`.

AWS CLI

Pour supprimer les paramètres de blocage de l'accès public d'un compte

L'`delete-public-access-block`exemple suivant supprime les paramètres de blocage de l'accès public pour le compte spécifié.

```
aws s3control delete-public-access-block \
    --account-id 123456789012
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePublicAccessBlockà la section Référence des AWS CLI commandes.

`describe-job`

L'exemple de code suivant montre comment utiliser`describe-job`.

AWS CLI

Pour décrire une tâche d'opérations par lots Amazon S3

Vous trouverez ci-dessous `describe-job` les paramètres de configuration et le statut de la tâche d'opérations par lots spécifiée.

```
aws s3control describe-job \
    --account-id 123456789012 \
    --job-id 93735294-df46-44d5-8638-6356f335324e
```

Sortie :

```
{
    "Job": {
        "TerminationDate": "2019-10-03T21:49:53.944Z",
        "JobId": "93735294-df46-44d5-8638-6356f335324e",
        "FailureReasons": [],
        "Manifest": {
            "Spec": {
                "Fields": [
                    "Bucket",
                    "Key"
                ],
                "Format": "S3BatchOperations_CSV_20180820"
            },
```

```
            "Location": {
                "ETag": "69f52a4e9f797e987155d9c8f5880897",
                "ObjectArn": "arn:aws:s3:::employee-records-logs/inv-
report/7a6a9be4-072c-407e-85a2-ec3e982f773e.csv"
            }
        },
        "Operation": {
            "S3PutObjectTagging": {
                "TagSet": [
                    {
                        "Value": "true",
                        "Key": "confidential"
                    }
                ]
            }
        },
        "RoleArn": "arn:aws:iam::123456789012:role/S3BatchJobRole",
        "ProgressSummary": {
            "TotalNumberOfTasks": 8,
            "NumberOfTasksFailed": 0,
            "NumberOfTasksSucceeded": 8
        },
        "Priority": 42,
        "Report": {
            "ReportScope": "AllTasks",
            "Format": "Report_CSV_20180820",
            "Enabled": true,
            "Prefix": "batch-op-create-job",
            "Bucket": "arn:aws:s3:::employee-records-logs"
        },
        "JobArn": "arn:aws:s3:us-west-2:123456789012:job/93735294-
df46-44d5-8638-6356f335324e",
        "CreationTime": "2019-10-03T21:48:48.048Z",
        "Status": "Complete"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeJobà la section Référence des AWS CLI commandes.

`get-access-point-policy-status`

L'exemple de code suivant montre comment utiliser`get-access-point-policy-status`.

AWS CLI

Pour récupérer l'état de la politique du point d'accès

L'`get-access-point-policy-status`exemple suivant récupère l'état de la politique de point d'accès pour le point d'accès nommé `finance-ap` dans le compte 123456789012. L'état de la politique du point d'accès indique si la politique du point d'accès autorise l'accès public. Avant d'exécuter cet exemple, remplacez le nom du point d'accès et le numéro de compte par des valeurs adaptées à votre cas d'utilisation.

```
aws s3control get-access-point-policy-status \
    --account-id 123456789012 \
    --name finance-ap
```

Sortie :

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

Pour plus d'informations sur les cas dans lesquels une politique de point d'accès est considérée comme publique, consultez La signification du terme « public » dans le manuel Amazon Simple Storage Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetAccessPointPolicyStatusà la section Référence des AWS CLI commandes.

`get-access-point-policy`

L'exemple de code suivant montre comment utiliser`get-access-point-policy`.

AWS CLI

Pour récupérer une politique de point d'accès

L'`get-access-point-policy`exemple suivant extrait la politique du point d'accès depuis le point d'accès nommé `finance-ap` dans le compte 123456789012. Avant d'exécuter cet exemple, remplacez le nom du point d'accès et le numéro de compte par des valeurs adaptées à votre cas d'utilisation.

```
aws s3control get-access-point-policy \
    --account-id 123456789012 \
    --name finance-ap
```

Sortie :

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/Admin\"},\"Action\":
\"s3:GetObject\",\"Resource\":\"arn:aws:s3:us-west-2:123456789012:accesspoint/
finance-ap/object/records/*\"}]}"
}
```

Pour plus d'informations, consultez la section Gestion de l'accès aux données avec les points d'accès Amazon S3 dans le guide du développeur Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous GetAccessPointPolicyà la section Référence des AWS CLI commandes.

`get-access-point`

L'exemple de code suivant montre comment utiliser`get-access-point`.

AWS CLI

Pour récupérer les détails de configuration du point d'accès

L'`get-access-point`exemple suivant extrait les détails de configuration du point d'accès nommé `finance-ap` dans le compte 123456789012. Avant d'exécuter cet exemple, remplacez le nom du point d'accès et le numéro de compte par des valeurs adaptées à votre cas d'utilisation.

```
aws s3control get-access-point \
    --account-id 123456789012 \
    --name finance-ap
```

Sortie :

```
{
    "Name": "finance-ap",
```

```
    "Bucket": "business-records",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": false,
        "IgnorePublicAcls": false,
        "BlockPublicPolicy": false,
        "RestrictPublicBuckets": false
    },
    "CreationDate": "2020-01-01T00:00:00Z"
}
```

Pour plus d'informations, consultez la section Gestion de l'accès aux données avec les points d'accès Amazon S3 dans le guide du développeur Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous GetAccessPointà la section Référence des AWS CLI commandes.

`get-multi-region-access-point-routes`

L'exemple de code suivant montre comment utiliser`get-multi-region-access-point-routes`.

AWS CLI

Pour demander la configuration actuelle de l'itinéraire du point d'accès multirégional

L'`get-multi-region-access-point-routes`exemple suivant renvoie la configuration de routage actuelle pour le point d'accès multirégional spécifié.

```
aws s3control get-multi-region-access-point-routes \
    --region Region \
    --account-id 111122223333 \
    --mrap MultiRegionAccessPoint_ARN
```

Sortie :

```
{
    "Mrap": "arn:aws:s3::111122223333:accesspoint/0000000000000.mrap",
    "Routes": [
        {
            "Bucket": "amzn-s3-demo-bucket1",
            "Region": "ap-southeast-2",
```

```
            "TrafficDialPercentage": 100
        },
        {
            "Bucket": "amzn-s3-demo-bucket2",
            "Region": "us-west-1",
            "TrafficDialPercentage": 0
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetMultiRegionAccessPointRoutesà la section Référence des AWS CLI commandes.

`get-public-access-block`

L'exemple de code suivant montre comment utiliser`get-public-access-block`.

AWS CLI

Pour répertorier les paramètres d'accès public bloqués pour un compte

L'`get-public-access-block`exemple suivant affiche les paramètres de blocage de l'accès public pour le compte spécifié.

```
aws s3control get-public-access-block \
    --account-id 123456789012
```

Sortie :

```
{
   "PublicAccessBlockConfiguration": {
      "BlockPublicPolicy": true,
      "RestrictPublicBuckets": true,
      "IgnorePublicAcls": true,
      "BlockPublicAcls": true
   }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPublicAccessBlockà la section Référence des AWS CLI commandes.

`list-access-points`

L'exemple de code suivant montre comment utiliser`list-access-points`.

AWS CLI

Exemple 1 : Pour récupérer la liste de tous les points d'accès d'un compte

L'`list-access-points`exemple suivant affiche une liste de tous les points d'accès associés aux buckets appartenant au compte 123456789012.

```
aws s3control list-access-points \
    --account-id 123456789012
```

Sortie :

```
{
    "AccessPointList": [
        {
            "Name": "finance-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "business-records"
        },
        {
            "Name": "managers-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "business-records"
        },
        {
            "Name": "private-network-ap",
            "NetworkOrigin": "VPC",
            "VpcConfiguration": {
                "VpcId": "1a2b3c"
            },
            "Bucket": "business-records"
        },
        {
            "Name": "customer-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        },
        {
            "Name": "public-ap",
```

```
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        }
    ]
}
```

Exemple 2 : pour récupérer la liste de tous les points d'accès d'un bucket

L'`list-access-points`exemple suivant extrait une liste de tous les points d'accès attachés au bucket `external-docs` appartenant au compte 123456789012.

```
aws s3control list-access-points \
    --account-id 123456789012 \
    --bucket external-docs
```

Sortie :

```
{
    "AccessPointList": [
        {
            "Name": "customer-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        },
        {
            "Name": "public-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion de l'accès aux données avec les points d'accès Amazon S3 dans le guide du développeur Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous ListAccessPointsà la section Référence des AWS CLI commandes.

`list-jobs`

L'exemple de code suivant montre comment utiliser`list-jobs`.

AWS CLI

Pour répertorier les tâches d'opérations par lots d'Amazon S3 d'un compte

L'`list-jobs`exemple suivant répertorie toutes les tâches d'opérations par lots récentes pour le compte spécifié.

```
aws s3control list-jobs \
    --account-id 123456789012
```

Sortie :

```
{
    "Jobs": [
        {
            "Operation": "S3PutObjectTagging",
            "ProgressSummary": {
                "NumberOfTasksFailed": 0,
                "NumberOfTasksSucceeded": 8,
                "TotalNumberOfTasks": 8
            },
            "CreationTime": "2019-10-03T21:48:48.048Z",
            "Status": "Complete",
            "JobId": "93735294-df46-44d5-8638-6356f335324e",
            "Priority": 42
        },
        {
            "Operation": "S3PutObjectTagging",
            "ProgressSummary": {
                "NumberOfTasksFailed": 0,
                "NumberOfTasksSucceeded": 0,
                "TotalNumberOfTasks": 0
            },
            "CreationTime": "2019-10-03T21:46:07.084Z",
            "Status": "Failed",
            "JobId": "3f3c7619-02d3-4779-97f6-1d98dd313108",
            "Priority": 42
        },
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListJobsà la section Référence des AWS CLI commandes.

put-access-point-policy

L'exemple de code suivant montre comment utiliserput-access-point-policy.

AWS CLI

Pour définir une politique de point d'accès

L'put-access-point-policyexemple suivant place la politique de point d'accès spécifiée pour le point d'accès finance-ap dans le compte 123456789012. Si le point d'accès dispose finance-ap déjà d'une politique, cette commande remplace la politique existante par celle spécifiée dans cette commande. Avant d'exécuter cet exemple, remplacez le numéro de compte, le nom du point d'accès et les déclarations de politique par des valeurs adaptées à votre cas d'utilisation.

```
aws s3control put-access-point-policy \
    --account-id 123456789012 \
    --name finance-ap \
    --policy file://ap-policy.json
```

Contenu de ap-policy.json :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Alice"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/finance-ap/
object/Alice/*"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de l'accès aux données avec les points d'accès Amazon S3 dans le guide du développeur Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous PutAccessPointPolicyà la section Référence des AWS CLI commandes.

`put-public-access-block`

L'exemple de code suivant montre comment utiliser`put-public-access-block`.

AWS CLI

Pour modifier les paramètres de blocage de l'accès public pour un compte

L'`put-public-access-block`exemple suivant fait passer tous les paramètres de blocage de l'accès public `true` au compte spécifié.

```
aws s3control put-public-access-block \
    --account-id 123456789012 \
    --public-access-block-configuration '{"BlockPublicAcls": true,
 "IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
 true}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutPublicAccessBlockà la section Référence des AWS CLI commandes.

`submit-multi-region-access-point-routes`

L'exemple de code suivant montre comment utiliser`submit-multi-region-access-point-routes`.

AWS CLI

Pour mettre à jour la configuration de routage de votre point d'accès multirégional

L'`submit-multi-region-access-point-routes`exemple suivant met à jour les états de routage de `amzn-s3-demo-bucket1` et `amzn-s3-demo-bucket2` dans la `ap-southeast-2` région pour votre point d'accès multirégional.

```
aws s3control submit-multi-region-access-point-routes \
    --region ap-southeast-2 \
```

```
    --account-id 111122223333 \
    --mrap MultiRegionAccessPoint_ARN \
    --route-updates Bucket=amzn-s3-demo-
bucket1,TrafficDialPercentage=100 Bucket=amzn-s3-demo-
bucket2,TrafficDialPercentage=0
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SubmitMultiRegionAccessPointRoutesà la section Référence des AWS CLI commandes.

`update-job-priority`

L'exemple de code suivant montre comment utiliser`update-job-priority`.

AWS CLI

Pour mettre à jour la priorité d'une tâche d'opérations par lots Amazon S3

L'`update-job-priority`exemple suivant met à jour la tâche spécifiée selon une nouvelle priorité.

```
aws s3control update-job-priority \
    --account-id 123456789012 \
    --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
    --priority 52
```

Sortie :

```
{
    "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",
    "Priority": 52
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateJobPriorityà la section Référence des AWS CLI commandes.

`update-job-status`

L'exemple de code suivant montre comment utiliser`update-job-status`.

AWS CLI

Pour mettre à jour le statut d'une tâche d'opérations par lots Amazon S3

L'`update-job-status`exemple suivant annule la tâche spécifiée en attente d'approbation.

```
aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
    --requested-job-status Cancelled
```

Sortie :

```
{
    "Status": "Cancelled",
    "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"
}
```

L'`update-job-status`exemple suivant confirme et exécute le paramètre spécifié en attente d'approbation.

```
aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \
    --requested-job-status Ready

Output::

{
    "Status": "Ready",
    "JobId": "5782949f-3301-4fb3-
be34-8d5bab54dbca"
}
```

L'`update-job-status`exemple suivant annule la tâche spécifiée en cours d'exécution.

```
 aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \
    --requested-job-status Cancelled

Output::
```

```
{
        "Status": "Cancelling",
        "JobId": "5782949f-3301-4fb3-be34-8d5bab54dbca"
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateJobStatusà la section Référence des AWS CLI commandes.

Exemples d'utilisation de S3 Glacier AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de S3 Glacier.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`abort-multipart-upload`

L'exemple de code suivant montre comment utiliser`abort-multipart-upload`.

AWS CLI

La commande suivante supprime un téléchargement partitionné en cours vers un coffre nommé : `my-vault`

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
 --upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

Cette commande ne génère pas de sortie. Amazon Glacier nécessite un argument d’ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d’union pour spécifier le compte

en cours d'utilisation. L'ID de téléchargement est renvoyé par la commande `aws glacier initiate-multipart-upload` et peut également être obtenu avec `aws glacier list-multipart-uploads`.

Pour plus d'informations sur les téléchargements partitionnés vers Amazon Glacier à l'aide de l'interface AWS de ligne de commande, consultez la section Utilisation d'Amazon Glacier dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous AbortMultipartUploadà la section Référence des AWS CLI commandes.

`abort-vault-lock`

L'exemple de code suivant montre comment utiliser`abort-vault-lock`.

AWS CLI

Pour abandonner un processus de verrouillage du coffre-fort en cours

L'`abort-vault-lock`exemple suivant supprime une politique de verrouillage du coffre-fort spécifié et rétablit l'état de verrouillage du coffre-fort sur déverrouillé.

```
aws glacier abort-vault-lock \
    --account-id - \
    --vault-name MyVaultName
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Abort Vault Lock (DELETE lock-policy) dans le guide du développeur d'API Amazon Glacier.

- Pour plus de détails sur l'API, reportez-vous AbortVaultLockà la section Référence des AWS CLI commandes.

`add-tags-to-vault`

L'exemple de code suivant montre comment utiliser`add-tags-to-vault`.

AWS CLI

La commande suivante ajoute deux balises à un coffre-fort nommé `my-vault` :

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --
tags id=1234,date=july2015
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous AddTagsToVaultà la section Référence des AWS CLI commandes.

`complete-multipart-upload`

L'exemple de code suivant montre comment utilisercomplete-multipart-upload.

AWS CLI

La commande suivante termine le téléchargement partitionné pour une archive de 3 Mio :

```
aws glacier complete-multipart-upload --archive-size 3145728 --
checksum 9628195fcdbcbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67
 --upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

L'ID de téléchargement est renvoyé par la commande `aws glacier initiate-multipart-upload` et peut également être obtenu avec `aws glacier list-multipart-uploads`. Le paramètre checksum prend un hachage de l'arborescence SHA-256 de l'archive en hexadécimal.

Pour plus d'informations sur les téléchargements partitionnés vers Amazon Glacier à l'aide de l' AWS interface de ligne de commande, y compris les instructions relatives au calcul d'un hachage arborescent, consultez la section Utilisation d'Amazon Glacier dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous CompleteMultipartUploadà la section Référence des AWS CLI commandes.

`complete-vault-lock`

L'exemple de code suivant montre comment utilisercomplete-vault-lock.

AWS CLI

Pour terminer un processus de verrouillage du coffre-fort en cours

L'`complete-vault-lock`exemple suivant complète la progression du verrouillage en cours pour le coffre-fort spécifié et définit l'état de verrouillage du coffre-fort sur`Locked`. Vous obtenez la valeur du `lock-id` paramètre lors de l'exécution`initiate-lock-process`.

```
aws glacier complete-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Complete Vault Lock (POST LockID) dans le guide du développeur de l'API Amazon Glacier.

- Pour plus de détails sur l'API, reportez-vous CompleteVaultLockà la section Référence des AWS CLI commandes.

`create-vault`

L'exemple de code suivant montre comment utiliser`create-vault`.

AWS CLI

La commande suivante crée un coffre-fort nommé `my-vault` :

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous CreateVaultà la section Référence des AWS CLI commandes.

`delete-archive`

L'exemple de code suivant montre comment utiliser`delete-archive`.

AWS CLI

Pour supprimer une archive d'un coffre-fort

L'exemple `delete-archive` suivant supprime l'archive spécifiée de `example_vault`.

```
aws glacier delete-archive \
    --account-id 111122223333  \
    --vault-name example_vault \
    --archive-id Sc0u9ZP8yaWkmh-XGlIvAVprtLhaLCGnNwNl5I5x9HqPIkX5mjc0DrId3Ln-
Gi_k2HzmlIDZUz117KSdVMdMXLuFWi9PJUitxWO73edQ43eTlMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteArchiveà la section Référence des AWS CLI commandes.

`delete-vault-access-policy`

L'exemple de code suivant montre comment utiliser`delete-vault-access-policy`.

AWS CLI

Pour supprimer la politique d'accès d'un coffre-fort

L'`delete-vault-access-policy`exemple suivant supprime la politique d'accès pour le coffre spécifié.

```
aws glacier delete-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteVaultAccessPolicyà la section Référence des AWS CLI commandes.

`delete-vault-notifications`

L'exemple de code suivant montre comment utiliser`delete-vault-notifications`.

AWS CLI

Pour supprimer les notifications SNS d'un coffre-fort

L'exemple `delete-vault-notifications` suivant supprime les notifications envoyées par Amazon Simple Notification Service (Amazon SNS) pour le coffre-fort spécifié.

```
aws glacier delete-vault-notifications \
    --account-id 111122223333 \
    --vault-name example_vault
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteVaultNotificationsà la section Référence des AWS CLI commandes.

`delete-vault`

L'exemple de code suivant montre comment utiliser`delete-vault`.

AWS CLI

La commande suivante supprime un coffre-fort nommé `my-vault` :

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

Cette commande ne génère pas de sortie. Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous DeleteVaultà la section Référence des AWS CLI commandes.

`describe-job`

L'exemple de code suivant montre comment utiliser`describe-job`.

AWS CLI

La commande suivante récupère des informations sur une tâche d'extraction d'inventaire sur un coffre-fort nommé `my-vault` :

```
aws glacier describe-job --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW
```

Sortie :

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "Completed": false,
    "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "Action": "InventoryRetrieval",
    "CreationDate": "2015-07-17T20:23:41.616Z",
    "StatusCode": "InProgress"
}
```

L'ID de tâche se trouve dans la sortie de `aws glacier initiate-job` et `aws glacier list-jobs`. Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous DescribeJobà la section Référence des AWS CLI commandes.

`describe-vault`

L'exemple de code suivant montre comment utiliser`describe-vault`.

AWS CLI

La commande suivante récupère les données relatives à un coffre-fort nommé `my-vault` :

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous DescribeVaultà la section Référence des AWS CLI commandes.

`get-data-retrieval-policy`

L'exemple de code suivant montre comment utiliser`get-data-retrieval-policy`.

AWS CLI

La commande suivante permet d'obtenir la politique de récupération des données pour le compte en cours d'utilisation :

```
aws glacier get-data-retrieval-policy --account-id -
```

Sortie :

```
{
    "Policy": {
        "Rules": [
            {
                "BytesPerHour": 10737418240,
                "Strategy": "BytesPerHour"
            }
        ]
    }
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous GetDataRetrievalPolicyà la section Référence des AWS CLI commandes.

`get-job-output`

L'exemple de code suivant montre comment utiliser`get-job-output`.

AWS CLI

La commande suivante enregistre la sortie d'une tâche d'inventaire du coffre-fort dans un fichier du répertoire actuel nommé `output.json` :

```
aws glacier get-job-output --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW output.json
```

Le `job-id` est disponible dans la sortie de `aws glacier list-jobs`. Notez que le nom du fichier de sortie est un argument positionnel qui n'est pas préfixé par un nom d'option. Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

Sortie :

```
{
    "status": 200,
    "acceptRanges": "bytes",
    "contentType": "application/json"
}
```

`output.json`:

```
{"VaultARN":"arn:aws:glacier:us-west-2:0123456789012:vaults/
my-vault","InventoryDate":"2015-04-07T00:26:18Z","ArchiveList":
[{"ArchiveId":"kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw","ArchiveDescription":"multipart
 upload
 test","CreationDate":"2015-04-06T22:24:34Z","Size":3145728,"SHA256TreeHash":"9628195fcdbcbb
```

- Pour plus de détails sur l'API, reportez-vous GetJobOutputà la section Référence des AWS CLI commandes.

`get-vault-access-policy`

L'exemple de code suivant montre comment utiliser`get-vault-access-policy`.

AWS CLI

Pour récupérer la politique d'accès d'un coffre

L'`get-vault-access-policy`exemple suivant récupère la politique d'accès pour le coffre spécifié.

```
aws glacier get-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

Sortie :

```
{
    "policy": {
        "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root
\"},\"Action\":\"glacier:ListJobs\",\"Resource\":\"arn:aws:glacier:us-
east-1:111122223333:vaults/example_vault\"},{\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":\"glacier:UploadArchive\",
\"Resource\":\"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}]}"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetVaultAccessPolicyà la section Référence des AWS CLI commandes.

`get-vault-lock`

L'exemple de code suivant montre comment utiliser`get-vault-lock`.

AWS CLI

Pour obtenir les détails d'un verrou de coffre-fort

L'`get-vault-lock`exemple suivant permet de récupérer les détails relatifs au verrou du coffre spécifié.

```
aws glacier get-vault-lock \
    --account-id - \
    --vault-name MyVaultName
```

Sortie :

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"Define-vault-
lock\",\"Effect\":\"Deny\",\"Principal\":{\"AWS\":\"arn:aws:iam::999999999999:root
\"},\"Action\":\"glacier:DeleteArchive\",\"Resource\":\"arn:aws:glacier:us-
```

```
west-2:99999999999:vaults/MyVaultName\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}",
    "State": "Locked",
    "CreationDate": "2019-07-29T22:25:28.640Z"
}
```

Pour plus d'informations, consultez Get Vault Lock (GET lock-policy) dans le guide du développeur d'API Amazon Glacier.

- Pour plus de détails sur l'API, reportez-vous GetVaultLockà la section Référence des AWS CLI commandes.

`get-vault-notifications`

L'exemple de code suivant montre comment utiliser`get-vault-notifications`.

AWS CLI

La commande suivante obtient une description de la configuration des notifications pour un coffre-fort nommé `my-vault` :

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

Sortie :

```
{
    "vaultNotificationConfig": {
        "Events": [
            "InventoryRetrievalCompleted",
            "ArchiveRetrievalCompleted"
        ],
        "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
    }
}
```

Si aucune notification n'a été configurée pour le coffre-fort, une erreur est renvoyée. Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous GetVaultNotificationsà la section Référence des AWS CLI commandes.

`initiate-job`

L'exemple de code suivant montre comment utiliser`initiate-job`.

AWS CLI

La commande suivante lance une tâche visant à obtenir un inventaire du coffre `my-vault` :

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
 '{"Type": "inventory-retrieval"}'
```

Sortie :

```
{
    "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW"
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

La commande suivante lance une tâche pour récupérer une archive depuis le coffre `my-vault` :

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

`job-archive-retrieval.json`est un fichier JSON situé dans le dossier local qui indique le type de tâche, l'ID d'archive et certains paramètres facultatifs :

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
  "Description": "Retrieve archive on 2015-07-17",
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

IDs Les archives sont disponibles dans la sortie de `aws glacier upload-archive` et`aws glacier get-job-output`.

Sortie :

```
{
    "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
    "jobId": "l7IL5-EkXy2O5uLYaFdAYOiEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav"
}
```

Consultez Initiate Job dans le manuel Amazon Glacier API Reference pour plus de détails sur le format des paramètres de tâche.

- Pour plus de détails sur l'API, reportez-vous InitiateJobà la section Référence des AWS CLI commandes.

`initiate-multipart-upload`

L'exemple de code suivant montre comment utiliser`initiate-multipart-upload`.

AWS CLI

La commande suivante lance un téléchargement partitionné vers un coffre-fort nommé `my-vault` avec une taille de pièce de 1 MiB (1024 x 1024 octets) par fichier :

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-
name my-vault --archive-description "multipart upload test"
```

Le paramètre de description de l'archive est facultatif. Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

Cette commande génère un ID de téléchargement en cas de réussite. Utilisez l'identifiant de téléchargement lorsque vous chargez chaque partie de votre archive avec`aws glacier upload-multipart-part`. Pour plus d'informations sur les téléchargements partitionnés vers Amazon Glacier à l'aide de l'interface AWS de ligne de commande, consultez la section Utilisation d'Amazon Glacier dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous InitiateMultipartUploadà la section Référence des AWS CLI commandes.

`initiate-vault-lock`

L'exemple de code suivant montre comment utiliser`initiate-vault-lock`.

AWS CLI

Pour lancer le processus de verrouillage du coffre-fort

L'`initiate-vault-lock`exemple suivant installe une politique de verrouillage du coffre-fort sur le coffre-fort spécifié et définit l'état de verrouillage du coffre-fort sur. `InProgress` Vous devez terminer le processus en appelant `complete-vault-lock` dans les 24 heures pour régler l'état du verrou du coffre-fort sur`Locked`.

```
aws glacier initiate-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --policy file://vault_lock_policy.json
```

Contenu de `vault_lock_policy.json` :

```
{"Policy":"{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"Define-vault-
lock\",\"Effect\":\"Deny\",\"Principal\":{\"AWS\":\"arn:aws:iam::999999999999:root
\"},\"Action\":\"glacier:DeleteArchive\",\"Resource\":\"arn:aws:glacier:us-
west-2:999999999999:vaults/examplevault\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}"}
```

Le résultat est l'ID de verrouillage du coffre-fort que vous pouvez utiliser pour terminer le processus de verrouillage du coffre-fort.

```
{
    "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"
}
```

Pour plus d'informations, consultez Initiate Vault Lock (POST lock-policy) dans le guide du développeur d'API Amazon Glacier.

- Pour plus de détails sur l'API, reportez-vous InitiateVaultLockà la section Référence des AWS CLI commandes.

`list-jobs`

L'exemple de code suivant montre comment utiliser`list-jobs`.

AWS CLI

La commande suivante répertorie les tâches en cours et récemment terminées pour un coffre-fort nommé `my-vault` :

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

Sortie :

```
{
    "JobList": [
        {
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "RetrievalByteRange": "0-3145727",
            "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
            "Completed": false,
            "SHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "JobId": "l7IL5-EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
            "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
            "JobDescription": "Retrieve archive on 2015-07-17",
            "ArchiveSizeInBytes": 3145728,
            "Action": "ArchiveRetrieval",
            "ArchiveSHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "CreationDate": "2015-07-17T21:16:13.840Z",
            "StatusCode": "InProgress"
        },
        {
            "InventoryRetrievalParameters": {
                "Format": "JSON"
            },
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "Completed": false,
            "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
```

```
            "Action": "InventoryRetrieval",
            "CreationDate": "2015-07-17T20:23:41.616Z",
            "StatusCode": ""InProgress""
        }
    ]
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous ListJobsà la section Référence des AWS CLI commandes.

`list-multipart-uploads`

L'exemple de code suivant montre comment utiliser`list-multipart-uploads`.

AWS CLI

La commande suivante affiche tous les téléchargements partitionnés en cours pour un coffre nommé : `my-vault`

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

Pour plus d'informations sur les téléchargements partitionnés vers Amazon Glacier à l'aide de l'interface AWS de ligne de commande, consultez la section Utilisation d'Amazon Glacier dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous ListMultipartUploadsà la section Référence des AWS CLI commandes.

`list-parts`

L'exemple de code suivant montre comment utiliser`list-parts`.

AWS CLI

La commande suivante répertorie les parties téléchargées pour un téléchargement en plusieurs parties dans un coffre nommé `my-vault` :

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-
YGqGwAm8Kn3BLP2ElNCvnB-5961R09CSaPmPwkYGHOqeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdrCs8ut-D"
```

Sortie :

```
{
    "MultipartUploadId": "SYZi7qnL-
YGqGwAm8Kn3BLP2ElNCvnB-5961R09CSaPmPwkYGHOqeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdrCs8ut-
D",
    "Parts": [
        {
            "RangeInBytes": "0-1048575",
            "SHA256TreeHash":
 "e1f2a7cd6e047350f69b9f8cfa60fa606fe2f02802097a9a026360a7edc1f553"
        },
        {
            "RangeInBytes": "1048576-2097151",
            "SHA256TreeHash":
 "43cf3061fb95796aed99a11a6aa3cd8f839eed15e655ab0a597126210636aee6"
        }
    ],
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "CreationDate": "2015-07-18T00:05:23.830Z",
    "PartSizeInBytes": 1048576
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

Pour plus d'informations sur les téléchargements partitionnés vers Amazon Glacier à l'aide de l'interface AWS de ligne de commande, consultez la section Utilisation d'Amazon Glacier dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous ListPartsà la section Référence des AWS CLI commandes.

`list-provisioned-capacity`

L'exemple de code suivant montre comment utiliser`list-provisioned-capacity`.

AWS CLI

Pour récupérer les unités de capacité provisionnées

L'`list-provisioned-capacity`exemple suivant permet de récupérer les détails de toutes les unités de capacité allouées pour le compte spécifié.

```
aws glacier list-provisioned-capacity \
    --account-id 111122223333
```

Sortie :

```
{
    "ProvisionedCapacityList": [
        {
            "CapacityId": "HpASAuvfRFiVDbOjMfEIcr8K",
            "ExpirationDate": "2020-03-18T19:59:24.000Z",
            "StartDate": "2020-02-18T19:59:24.912Z"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListProvisionedCapacityà la section Référence des AWS CLI commandes.

`list-tags-for-vault`

L'exemple de code suivant montre comment utiliser`list-tags-for-vault`.

AWS CLI

La commande suivante répertorie les balises appliquées à un coffre-fort nommé `my-vault` :

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

Sortie :

```
{
    "Tags": {
        "date": "july2015",
        "id": "1234"
    }
```

```
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous ListTagsForVaultà la section Référence des AWS CLI commandes.

`list-vaults`

L'exemple de code suivant montre comment utiliser`list-vaults`.

AWS CLI

La commande suivante répertorie les coffres-forts dans le compte et la région par défaut :

```
aws glacier list-vaults --account-id -
```

Sortie :

```
{
    "VaultList": [
        {
            "SizeInBytes": 3178496,
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "LastInventoryDate": "2015-04-07T00:26:19.028Z",
            "VaultName": "my-vault",
            "NumberOfArchives": 1,
            "CreationDate": "2015-04-06T21:23:45.708Z"
        }
    ]
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous ListVaultsà la section Référence des AWS CLI commandes.

`purchase-provisioned-capacity`

L'exemple de code suivant montre comment utiliser`purchase-provisioned-capacity`.

AWS CLI

Pour acheter une unité de capacité provisionnée

L'`purchase-provisioned-capacity`exemple suivant achète une unité de capacité provisionnée.

```
aws glacier purchase-provisioned-capacity \
    --account-id 111122223333
```

Sortie :

```
{
    "capacityId": "HpASAuvfRFiVDbOjMfEIcr8K"
}
```

- Pour plus de détails sur l'API, reportez-vous PurchaseProvisionedCapacityà la section Référence des AWS CLI commandes.

`remove-tags-from-vault`

L'exemple de code suivant montre comment utiliser`remove-tags-from-vault`.

AWS CLI

La commande suivante supprime une balise contenant la clé `date` d'un coffre nommé `my-vault` :

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-keys date
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous RemoveTagsFromVaultà la section Référence des AWS CLI commandes.

`set-data-retrieval-policy`

L'exemple de code suivant montre comment utiliser`set-data-retrieval-policy`.

AWS CLI

La commande suivante configure une politique de récupération de données pour le compte en cours d'utilisation :

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-policy.json
```

`data-retrieval-policy.json`est un fichier JSON dans le dossier actuel qui spécifie une politique de récupération des données :

```
{
  "Rules":[
     {
         "Strategy":"BytesPerHour",
         "BytesPerHour":10737418240
      }
   ]
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

La commande suivante définit la politique de récupération des données de manière à `FreeTier` utiliser le JSON intégré :

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules":[{"Strategy":"FreeTier"}]}'
```

Consultez la section Définir la politique de récupération des données dans le manuel Amazon Glacier API Reference pour plus de détails sur le format de la politique.

- Pour plus de détails sur l'API, reportez-vous SetDataRetrievalPolicyà la section Référence des AWS CLI commandes.

`set-vault-access-policy`

L'exemple de code suivant montre comment utiliser`set-vault-access-policy`.

AWS CLI

Pour définir la politique d'accès d'un coffre-fort

L'`set-vault-access-policy`exemple suivant attache une politique d'autorisation au coffre spécifié.

```
aws glacier set-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
    --policy '{"Policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root
\"},\"Action\":\"glacier:ListJobs\",\"Resource\":\"arn:aws:glacier:us-
east-1:111122223333:vaults/example_vault\"},{\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":\"glacier:UploadArchive\",
\"Resource\":\"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}]}"}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetVaultAccessPolicyà la section Référence des AWS CLI commandes.

`set-vault-notifications`

L'exemple de code suivant montre comment utiliser`set-vault-notifications`.

AWS CLI

La commande suivante configure les notifications SNS pour un coffre-fort nommé `my-vault` :

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-
notification-config file://notificationconfig.json
```

`notificationconfig.json` est un fichier JSON dans le dossier actuel qui indique une rubrique SNS et les événements à publier :

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

- Pour plus de détails sur l'API, reportez-vous SetVaultNotificationsà la section Référence des AWS CLI commandes.

`upload-archive`

L'exemple de code suivant montre comment utiliser`upload-archive`.

AWS CLI

La commande suivante charge une archive du dossier actuel nommé `archive.zip` vers un coffre-fort nommé `my-vault` :

```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

Sortie :

```
{
    "archiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
    "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
    "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

Pour récupérer une archive chargée, lancez une tâche d'extraction à l'aide de la commande aws glacier initiate-job.

- Pour plus de détails sur l'API, reportez-vous UploadArchiveà la section Référence des AWS CLI commandes.

`upload-multipart-part`

L'exemple de code suivant montre comment utiliser`upload-multipart-part`.

AWS CLI

La commande suivante télécharge le premier Mio (1 024 x 1 024 octets) d'une archive :

```
aws glacier upload-multipart-part --body part1 --range 'bytes
 0-1048575/*' --account-id - --vault-name my-vault --upload-
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

Amazon Glacier nécessite un argument d'ID de compte pour effectuer des opérations, mais vous pouvez utiliser un trait d'union pour spécifier le compte en cours d'utilisation.

Le paramètre corps indique le chemin d'un fichier partiel sur le système de fichiers local. Le paramètre plage prend une plage de contenu HTTP indiquant les octets que la partie occupe dans l'archive terminée. L'ID de téléchargement est renvoyé par la commande `aws glacier initiate-multipart-upload` et peut également être obtenu avec `aws glacier list-multipart-uploads`.

Pour plus d'informations sur les téléchargements partitionnés vers Amazon Glacier à l'aide de l'interface AWS de ligne de commande, consultez la section Utilisation d'Amazon Glacier dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous UploadMultipartPartà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Secrets Manager AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with Secrets Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-get-secret-value`

L'exemple de code suivant montre comment utiliser`batch-get-secret-value`.

AWS CLI

Exemple 1 : Pour récupérer la valeur secrète d'un groupe de secrets listés par nom

L'`batch-get-secret-value`exemple suivant obtient la valeur secrète secrets pour trois secrets.

```
aws secretsmanager batch-get-secret-value \
    --secret-id-list MySecret1 MySecret2 MySecret3
```

Sortie :

```
{
    "SecretValues": [
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
            "Name": "MySecret1",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
            "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1523477145.729"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
            "Name": "MySecret2",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
```

```
            ],
            "CreatedDate": "1673477781.275"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
            "Name": "MySecret3",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEccccc",
            "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1373477721.124"
        }
    ],
    "Errors": []
}
```

Pour plus d'informations, consultez la section Récupérer un groupe de secrets dans un lot dans le Guide de l'utilisateur de AWS Secrets Manager.

Exemple 2 : Pour récupérer la valeur secrète d'un groupe de secrets sélectionnés par filtre

L'`batch-get-secret-value`exemple suivant permet d'obtenir la valeur secrète (secrets) de votre compte dont le nom `MySecret` figure dans le nom. Le filtrage par nom est sensible à la casse.

```
aws secretsmanager batch-get-secret-value \
    --filters Key="name",Values="MySecret"
```

Sortie :

```
{
    "SecretValues": [
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
            "Name": "MySecret1",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
            "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
```

```
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1523477145.729"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
            "Name": "MySecret2",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1673477781.275"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
            "Name": "MySecret3",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEccccc",
            "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1373477721.124"
        }
    ],
    "Errors": []
}
```

Pour plus d'informations, consultez la section Récupérer un groupe de secrets dans un lot dans le Guide de l'utilisateur de AWS Secrets Manager.

- Pour plus de détails sur l'API, voir BatchGetSecretValuela section Référence des AWS CLI commandes.

`cancel-rotate-secret`

L'exemple de code suivant montre comment utiliser`cancel-rotate-secret`.

AWS CLI

Pour désactiver la rotation automatique pour un secret

L'`cancel-rotate-secret`exemple suivant désactive la rotation automatique pour un secret. Pour reprendre la rotation, appelez`rotate-secret`.

```
aws secretsmanager cancel-rotate-secret \
    --secret-id MyTestSecret
```

Sortie :

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret"
}
```

Pour plus d'informations, voir Rotation d'un secret dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir CancelRotateSecretla section Référence des AWS CLI commandes.

`create-secret`

L'exemple de code suivant montre comment utiliser`create-secret`.

AWS CLI

Exemple 1 : pour créer un secret à partir des informations d'identification d'un fichier JSON

L'exemple suivant `create-secret` crée un secret à partir des informations d'identification d'un fichier. Pour plus d'informations, consultez la section Chargement des paramètres de la AWS CLI à partir d'un fichier dans le Guide de l'utilisateur de la AWS CLI.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
```

```
    --secret-string file://mycreds.json
```

Contenu de `mycreds.json` :

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

Sortie :

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Création d'un secret dans le guide de l'utilisateur de Secrets Manager.

Exemple 2 : pour créer un secret

L'exemple suivant `create-secret` crée un secret avec deux paires clé-valeur. Lorsque saisissez des commandes dans un shell de commande, il existe un risque d'accès à l'historique des commandes ou aux paramètres de vos commandes. Cela pose problème si la commande inclut la valeur d'un secret. Pour plus d'informations, consultez la section Atténuer les risques liés à l'utilisation d'outils de ligne de commande pour stocker des secrets dans le Guide de l'utilisateur de Secrets Manager.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --description "My test secret created with the CLI." \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

Sortie :

```
{
```

```
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"
}
```

Pour plus d'informations, consultez la section Création d'un secret dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir CreateSecretla section Référence des AWS CLI commandes.

`delete-resource-policy`

L'exemple de code suivant montre comment utiliser`delete-resource-policy`.

AWS CLI

Pour supprimer la politique basée sur les ressources attachée à un secret

L'exemple suivant `delete-resource-policy` supprime la politique basée sur les ressources associée à un secret.

```
aws secretsmanager delete-resource-policy \
    --secret-id MyTestSecret
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, consultez Authentification et contrôle d'accès dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir DeleteResourcePolicyla section Référence des AWS CLI commandes.

`delete-secret`

L'exemple de code suivant montre comment utiliser`delete-secret`.

AWS CLI

Exemple 1 : pour supprimer un secret

L'exemple suivant `delete-secret` supprime un secret répliqué. Vous pouvez récupérer le secret `restore-secret` jusqu'à la date et à l'heure indiquées dans le champ de `DeletionDate` réponse. Pour supprimer un secret répliqué dans d'autres régions, supprimez d'abord ses répliques avec `remove-regions-from-replication`, puis appelez `delete-secret`.

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --recovery-window-in-days 7
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "DeletionDate": 1524085349.095
}
```

Pour plus d'informations, voir Supprimer un secret dans le Guide de l'utilisateur de Secrets Manager.

Exemple 2 : Supprimer un secret immédiatement

L'exemple suivant `delete-secret` supprime immédiatement le secret, sans plage de récupération. Il n'est pas possible de récupérer ce secret.

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --force-delete-without-recovery
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "DeletionDate": 1508750180.309
```

```
}
```

Pour plus d'informations, voir Supprimer un secret dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir DeleteSecretla section Référence des AWS CLI commandes.

`describe-secret`

L'exemple de code suivant montre comment utiliser`describe-secret`.

AWS CLI

Pour récupérer les détails d'un secret

L'`describe-secret`exemple suivant montre les détails d'un secret.

```
aws secretsmanager describe-secret \
    --secret-id MyTestSecret
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
Ca8JGt",
    "Name": "MyTestSecret",
    "Description": "My test secret",
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-
ba987EXAMPLE",
    "RotationEnabled": true,
    "RotationLambdaARN": "arn:aws:lambda:us-
west-2:123456789012:function:MyTestRotationLambda",
    "RotationRules": {
        "AutomaticallyAfterDays": 2,
        "Duration": "2h",
        "ScheduleExpression": "cron(0 16 1,15 * ? *)"
    },
    "LastRotatedDate": 1525747253.72,
    "LastChangedDate": 1523477145.729,
    "LastAccessedDate": 1524572133.25,
    "Tags": [
        {
            "Key": "SecondTag",
```

```
            "Value": "AnotherValue"
        },
        {
            "Key": "FirstTag",
            "Value": "SomeValue"
        }
    ],
    "VersionIdsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
            "AWSPREVIOUS"
        ],
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
            "AWSCURRENT"
        ],
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333": [
            "AWSPENDING"
        ]
    },
    "CreatedDate": 1521534252.66,
    "PrimaryRegion": "us-west-2",
    "ReplicationStatus": [
        {
            "Region": "eu-west-3",
            "KmsKeyId": "alias/aws/secretsmanager",
            "Status": "InSync",
            "StatusMessage": "Replication succeeded"
        }
    ]
}
```

Pour plus d'informations, consultez la section Secret dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir DescribeSecretla section Référence des AWS CLI commandes.

`get-random-password`

L'exemple de code suivant montre comment utiliser`get-random-password`.

AWS CLI

Pour générer un mot de passe aléatoire

L'`get-random-password`exemple suivant génère un mot de passe aléatoire de 20 caractères comprenant au moins une lettre majuscule, une lettre minuscule, un chiffre et un signe de ponctuation.

```
aws secretsmanager get-random-password \
    --require-each-included-type \
    --password-length 20
```

Sortie :

```
{
    "RandomPassword": "EXAMPLE-PASSWORD"
}
```

Pour plus d'informations, consultez la section Création et gestion de secrets dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir GetRandomPasswordla section Référence des AWS CLI commandes.

`get-resource-policy`

L'exemple de code suivant montre comment utiliser`get-resource-policy`.

AWS CLI

Pour récupérer la politique basée sur les ressources attachée à un secret

L'exemple suivant `get-resource-policy` récupère la politique basée sur les ressources associée à un secret.

```
aws secretsmanager get-resource-policy \
    --secret-id MyTestSecret
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
```

```
    "ResourcePolicy": "{\n\"Version\":\"2012-10-17\",\n\"Statement\":[{\n\"Effect\":
\"Allow\",\n
    \"Principal\":{\n\"AWS\":\"arn:aws:iam::123456789012:root\"\n},\n\"Action\":
    \"secretsmanager:GetSecretValue\",\n\"Resource\":\"*\"\n}]\n}"
}
```

Pour plus d'informations, consultez Authentification et contrôle d'accès dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir GetResourcePolicyla section Référence des AWS CLI commandes.

`get-secret-value`

L'exemple de code suivant montre comment utiliser`get-secret-value`.

AWS CLI

Exemple 1 : Pour récupérer la valeur secrète cryptée d'un secret

L'exemple suivant `get-secret-value` récupère la valeur actuelle du secret.

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SecretString": "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": 1523477145.713
}
```

Pour plus d'informations, consultez la section Récupérer un secret dans le guide de l'utilisateur de Secrets Manager.

Exemple 2 : Pour récupérer la valeur secrète précédente

L'`get-secret-value`exemple suivant obtient la valeur secrète précédente. :

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
    --version-stage AWSPREVIOUS
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "SecretString": "{\"user\":\"diegor\",\"password\":\"PREVIOUS-EXAMPLE-PASSWORD
\"}",
    "VersionStages": [
        "AWSPREVIOUS"
    ],
    "CreatedDate": 1523477145.713
}
```

Pour plus d'informations, consultez la section Récupérer un secret dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir GetSecretValuela section Référence des AWS CLI commandes.

`list-secret-version-ids`

L'exemple de code suivant montre comment utiliser`list-secret-version-ids`.

AWS CLI

Pour répertorier toutes les versions secrètes associées à un secret

L'`list-secret-version-ids`exemple suivant permet d'obtenir une liste de toutes les versions d'un secret.

```
aws secretsmanager list-secret-version-ids \
    --secret-id MyTestSecret
```

Sortie :

```
{
  "Versions": [
    {
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "VersionStages": [
            "AWSPREVIOUS"
        ],
        "LastAccessedDate": 1523477145.713,
        "CreatedDate": 1523477145.713
    },
    {
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "VersionStages": [
            "AWSCURRENT"
        ],
        "LastAccessedDate": 1523477145.713,
        "CreatedDate": 1523486221.391
    },
    {
        "CreatedDate": 1.51197446236E9,
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333;"
    }
    ],
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, consultez la section Version dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir ListSecretVersionIdsla section Référence des AWS CLI commandes.

`list-secrets`

L'exemple de code suivant montre comment utiliser`list-secrets`.

AWS CLI

Exemple 1 : Pour répertorier les secrets de votre compte

L'exemple suivant `list-secrets` permet d'obtenir la liste des secrets de votre compte.

```
aws secretsmanager list-secrets
```

Sortie :

```
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
            "Name": "MyTestSecret",
            "LastChangedDate": 1523477145.729,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
                    "AWSCURRENT"
                ]
            }
        },
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:AnotherSecret-d4e5f6",
            "Name": "AnotherSecret",
            "LastChangedDate": 1523482025.685,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
                    "AWSCURRENT"
                ]
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Rechercher un secret dans le guide de l'utilisateur de Secrets Manager.

Exemple 2 : Pour filtrer la liste des secrets de votre compte

L'`list-secrets`exemple suivant permet d'obtenir une liste des secrets de votre compte dont le nom figure `Test` dans le nom. Le filtrage par nom est sensible à la casse.

```
aws secretsmanager list-secrets \
```

```
    --filter Key="name",Values="Test"
```

Sortie :

```
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
            "Name": "MyTestSecret",
            "LastChangedDate": 1523477145.729,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
                    "AWSCURRENT"
                ]
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Rechercher un secret dans le guide de l'utilisateur de Secrets Manager.

Exemple 3 : Pour répertorier les secrets de votre compte géré par un autre service

L'`list-secrets`exemple suivant renvoie les secrets de votre compte qui sont gérés par Amazon RDS.

```
aws secretsmanager list-secrets \
    --filter Key="owning-service",Values="rds"
```

Sortie :

```
{
    "SecretList": [
        {
            "Name": "rds!cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Tags": [
                {
                    "Value": "arn:aws:rds:us-
west-2:123456789012:cluster:database-1",
```

```
                    "Key": "aws:rds:primaryDBClusterArn"
                },
                {
                    "Value": "rds",
                    "Key": "aws:secretsmanager:owningService"
                }
            ],
            "RotationRules": {
                "AutomaticallyAfterDays": 1
            },
            "LastChangedDate": 1673477781.275,
            "LastRotatedDate": 1673477781.26,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa": [
                    "AWSPREVIOUS"
                ],
                "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb": [
                    "AWSCURRENT",
                    "AWSPENDING"
                ]
            },
            "OwningService": "rds",
            "RotationEnabled": true,
            "CreatedDate": 1673467300.7,
            "LastAccessedDate": 1673395200.0,
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:rds!
cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-a1b2c3",
            "Description": "Secret associated with primary RDS DB cluster:
 arn:aws:rds:us-west-2:123456789012:cluster:database-1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Secrets gérés par d'autres services dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir ListSecretsla section Référence des AWS CLI commandes.

`put-resource-policy`

L'exemple de code suivant montre comment utiliser`put-resource-policy`.

AWS CLI

Pour ajouter une politique basée sur les ressources à un secret

L'exemple suivant `put-resource-policy` ajoute une politique d'autorisations à un secret, en vérifiant d'abord que la politique ne fournit pas un accès étendu au secret. La politique est lue à partir d'un fichier. Pour plus d'informations, consultez la section Chargement des paramètres de la AWS CLI à partir d'un fichier dans le Guide de l'utilisateur de la AWS CLI.

```
aws secretsmanager put-resource-policy \
    --secret-id MyTestSecret \
    --resource-policy file://mypolicy.json \
    --block-public-policy
```

Contenu de `mypolicy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, voir Associer une politique d'autorisation à un secret dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir PutResourcePolicyla section Référence des AWS CLI commandes.

`put-secret-value`

L'exemple de code suivant montre comment utiliser`put-secret-value`.

AWS CLI

Exemple 1 : Pour stocker une nouvelle valeur secrète dans un secret

L'`put-secret-value`exemple suivant crée une nouvelle version d'un secret avec deux paires clé-valeur.

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "VersionStages": [
        "AWSCURRENT"
    ]
}
```

Pour plus d'informations, voir Modifier un secret dans le Guide de l'utilisateur de Secrets Manager.

Exemple 2 : pour stocker une nouvelle valeur secrète à partir des informations d'identification dans un fichier JSON

L'exemple suivant `put-secret-value` crée une nouvelle version d'un secret à partir des informations d'identification d'un fichier. Pour plus d'informations, consultez la section Chargement des paramètres de la AWS CLI à partir d'un fichier dans le Guide de l'utilisateur de la AWS CLI.

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
```

```
    --secret-string file://mycreds.json
```

Contenu de `mycreds.json` :

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "VersionStages": [
        "AWSCURRENT"
    ]
}
```

Pour plus d'informations, voir Modifier un secret dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir PutSecretValuela section Référence des AWS CLI commandes.

`remove-regions-from-replication`

L'exemple de code suivant montre comment utiliser`remove-regions-from-replication`.

AWS CLI

Pour supprimer un secret de réplique

L'exemple suivant `remove-regions-from-replication` supprime un secret répliqué dans eu-west-3. Pour supprimer un secret principal répliqué dans d'autres régions, supprimez d'abord les répliques, puis appelez `delete-secret`.

```
aws secretsmanager remove-regions-from-replication \
    --secret-id MyTestSecret \
    --remove-replica-regions eu-west-3
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "ReplicationStatus": []
}
```

Pour plus d'informations, consultez la section Supprimer une réplique secrète dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir RemoveRegionsFromReplicationla section Référence des AWS CLI commandes.

`replicate-secret-to-regions`

L'exemple de code suivant montre comment utiliser`replicate-secret-to-regions`.

AWS CLI

Pour reproduire un secret dans une autre région

L'exemple suivant `replicate-secret-to-regions` réplique un secret vers la zone eu-west-3. La réplique est chiffrée à l'aide de la clé AWS géréeaws/secretsmanager.

```
aws secretsmanager replicate-secret-to-regions \
    --secret-id MyTestSecret \
    --add-replica-regions Region=eu-west-3
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "ReplicationStatus": [
        {
```

```
            "Region": "eu-west-3",
            "KmsKeyId": "alias/aws/secretsmanager",
            "Status": "InProgress"
        }
    ]
}
```

Pour plus d'informations, consultez la section Répliquer un secret dans une autre région dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir ReplicateSecretToRegionsla section Référence des AWS CLI commandes.

`restore-secret`

L'exemple de code suivant montre comment utiliser`restore-secret`.

AWS CLI

Pour restaurer un secret précédemment supprimé

L'exemple suivant `restore-secret` restaure un secret dont la suppression était précédemment planifiée.

```
aws secretsmanager restore-secret \
    --secret-id MyTestSecret
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, voir Supprimer un secret dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir RestoreSecretla section Référence des AWS CLI commandes.

`rotate-secret`

L'exemple de code suivant montre comment utiliser`rotate-secret`.

AWS CLI

Exemple 1 : pour configurer et démarrer la rotation automatique d'un secret

L'`rotate-secret`exemple suivant configure et lance la rotation automatique d'un secret. Secrets Manager fait pivoter le secret une fois immédiatement, puis toutes les huit heures sur une période de deux heures. La sortie indique `VersionId` la nouvelle version secrète créée par rotation.

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret \
    --rotation-lambda-arn arn:aws:lambda:us-
west-2:1234566789012:function:SecretsManagerTestRotationLambda \
    --rotation-rules "{\"ScheduleExpression\": \"cron(0 8/8 * * ? *)\", \"Duration
\": \"2h\"}"
```

Sortie :

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez Rotation des secrets dans le Guide de l'utilisateur de Secrets Manager.

Exemple 2 : pour configurer et démarrer la rotation automatique sur un intervalle de rotation

L'`rotate-secret`exemple suivant configure et lance la rotation automatique d'un secret. Secrets Manager fait pivoter le secret une fois immédiatement, puis tous les 10 jours. La sortie indique `VersionId` la nouvelle version secrète créée par rotation.

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret \
    --rotation-lambda-arn arn:aws:lambda:us-
west-2:1234566789012:function:SecretsManagerTestRotationLambda \
    --rotation-rules "{\"ScheduleExpression\": \"rate(10 days)\"}"
```

Sortie :

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez Rotation des secrets dans le Guide de l'utilisateur de Secrets Manager.

Exemple 3 : pour faire pivoter un secret immédiatement

L'exemple suivant `rotate-secret` lance une rotation immédiate. La sortie indique `VersionId` la nouvelle version secrète créée par rotation. La rotation du secret doit déjà être configurée à l'heure actuelle.

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret
```

Sortie :

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez Rotation des secrets dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir RotateSecretla section Référence des AWS CLI commandes.

`stop-replication-to-replica`

L'exemple de code suivant montre comment utiliser`stop-replication-to-replica`.

AWS CLI

Pour promouvoir une réplique de secret auprès d'un serveur principal

L'exemple suivant `stop-replication-to-replica` supprime le lien entre un secret de réplique et le secret principal. Le secret répliqué est promu en secret principal dans la région de la réplique. Vous devez appeler `stop-replication-to-replica` depuis la région où se trouve la réplique.

```
aws secretsmanager stop-replication-to-replica \
    --secret-id MyTestSecret
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3"
}
```

Pour plus d'informations, consultez la section Promouvoir une réplique secrète dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir StopReplicationToReplicala section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Exemple 1 : pour ajouter un tag à un secret

L'exemple suivant montre comment associer une identification à l'aide d'une syntaxe abrégée.

```
aws secretsmanager tag-resource \
    --secret-id MyTestSecret \
    --tags Key=FirstTag,Value=FirstValue
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquer vos secrets dans le guide de l'utilisateur de Secrets Manager.

Exemple 2 : pour ajouter plusieurs balises à un secret

L'exemple suivant `tag-resource` associe deux balises clé-valeur à un secret.

```
aws secretsmanager tag-resource \
    --secret-id MyTestSecret \
    --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",
 "Value": "SecondValue"}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Tag secrets dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir TagResourcela section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer les tags d'un secret

L'exemple suivant `untag-resource` montre comment supprimer deux identifications d'un secret. Pour chaque identification, la clé et la valeur sont toutes les deux supprimées.

```
aws secretsmanager untag-resource \
    --secret-id MyTestSecret \
    --tag-keys '[ "FirstTag", "SecondTag"]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Tag secrets dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir UntagResourcela section Référence des AWS CLI commandes.

`update-secret-version-stage`

L'exemple de code suivant montre comment utiliser`update-secret-version-stage`.

AWS CLI

Exemple 1 : pour rétablir un secret dans la version précédente

L'`update-secret-version-stage`exemple suivant déplace l'étiquette de mise en scène AWS CURRENT vers la version précédente d'un secret, ce qui rétablit le secret dans sa version précédente. Pour trouver l'ID de la version précédente, utilisez`list-secret-version-ids`. Dans cet exemple, la version portant l'étiquette AWS CURRENT est a1b2c3d4-5678-90ab-cdef- et la version portant l'étiquette PREVIOUS est a1b2c3d4-5678-90ab-cdef-. EXAMPLE11111 AWS EXAMPLE22222 Dans cet exemple, vous déplacez l'étiquette AWS CURRENT de la version 11111 à la version 22222. L'étiquette AWS ACTUELLE étant supprimée d'une version, l'étiquette AWS PRÉCÉDENTE est `update-secret-version-stage` automatiquement déplacée vers cette version (11111). L'effet est que les versions AWS ACTUELLE et AWS PRÉCÉDENTE sont échangées.

```
aws secretsmanager update-secret-version-stage \
    --secret-id MyTestSecret \
    --version-stage AWSCURRENT \
    --move-to-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
    --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, consultez la section Version dans le guide de l'utilisateur de Secrets Manager.

Exemple 2 : pour ajouter une étiquette intermédiaire attachée à une version d'un secret

L'`update-secret-version-stage`exemple suivant ajoute une étiquette intermédiaire à une version d'un secret. Vous pouvez consulter les résultats en exécutant `list-secret-version-ids` et en affichant le champ de `VersionStages` réponse correspondant à la version concernée.

```
aws secretsmanager update-secret-version-stage \
```

```
    --secret-id MyTestSecret \
    --version-stage STAGINGLABEL1 \
    --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, consultez la section Version dans le guide de l'utilisateur de Secrets Manager.

Exemple 3 : pour supprimer une étiquette intermédiaire attachée à une version d'un secret

L'`update-secret-version-stage`exemple suivant supprime une étiquette intermédiaire attachée à une version d'un secret. Vous pouvez consulter les résultats en exécutant `list-secret-version-ids` et en affichant le champ de `VersionStages` réponse correspondant à la version concernée.

```
aws secretsmanager update-secret-version-stage \
    --secret-id MyTestSecret \
    --version-stage STAGINGLABEL1 \
    --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, consultez la section Version dans le guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir UpdateSecretVersionStagela section Référence des AWS CLI commandes.

`update-secret`

L'exemple de code suivant montre comment utiliser`update-secret`.

AWS CLI

Exemple 1 : pour mettre à jour la description d'un secret

L'exemple suivant `update-secret` met à jour la description d'un secret.

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --description "This is a new description for the secret."
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, voir Modifier un secret dans le Guide de l'utilisateur de Secrets Manager.

Exemple 2 : pour mettre à jour la clé de chiffrement associée à un secret

L'exemple suivant `update-secret` met à jour la clé KMS utilisée pour chiffrer la valeur secrète. Les clés KMS doivent être situées dans la même région que le secret.

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-
ba987EXAMPLE
```

Sortie :

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

Pour plus d'informations, voir Modifier un secret dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir UpdateSecretla section Référence des AWS CLI commandes.

`validate-resource-policy`

L'exemple de code suivant montre comment utiliser`validate-resource-policy`.

AWS CLI

Pour valider une politique de ressources

L'`validate-resource-policy`exemple suivant vérifie qu'une politique de ressources n'accorde pas un accès étendu à un secret. La politique est lue à partir d'un fichier sur le disque. Pour plus d'informations, consultez la section Chargement des paramètres de la AWS CLI à partir d'un fichier dans le Guide de l'utilisateur de la AWS CLI.

```
aws secretsmanager validate-resource-policy \
    --resource-policy file://mypolicy.json
```

Contenu de `mypolicy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

Sortie :

```
{
    "PolicyValidationPassed": true,
    "ValidationErrors": []
```

```
}
```

Pour plus d'informations, reportez-vous à la section Référence des autorisations pour Secrets Manager dans le Guide de l'utilisateur de Secrets Manager.

- Pour plus de détails sur l'API, voir ValidateResourcePolicyla section Référence des AWS CLI commandes.

Exemples d'utilisation de Security Hub AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Security Hub.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-administrator-invitation`

L'exemple de code suivant montre comment utiliser`accept-administrator-invitation`.

AWS CLI

Pour accepter une invitation provenant d'un compte administrateur

L'`accept-administrator-invitation`exemple suivant accepte l'invitation spécifiée provenant du compte administrateur spécifié.

```
aws securityhub accept-invitation \
    --administrator-id 123456789012 \
    --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous AcceptAdministratorInvitationà la section Référence des AWS CLI commandes.

`accept-invitation`

L'exemple de code suivant montre comment utiliser`accept-invitation`.

AWS CLI

Pour accepter une invitation provenant d'un compte administrateur

L'`accept-invitation`exemple suivant accepte l'invitation spécifiée provenant du compte administrateur spécifié.

```
aws securityhub accept-invitation \
    --master-id 123456789012 \
    --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous AcceptInvitationà la section Référence des AWS CLI commandes.

`batch-delete-automation-rules`

L'exemple de code suivant montre comment utiliser`batch-delete-automation-rules`.

AWS CLI

Pour supprimer des règles d'automatisation

L'`batch-delete-automation-rules`exemple suivant supprime la règle d'automatisation spécifiée. Vous pouvez supprimer une ou plusieurs règles à l'aide d'une seule commande. Seul le compte administrateur du Security Hub peut exécuter cette commande.

```
aws securityhub batch-delete-automation-rules \
```

```
    --automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

Sortie :

```
{
    "ProcessedAutomationRules": [
        "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
}
```

Pour plus d'informations, consultez la section Suppression des règles d'automatisation dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchDeleteAutomationRulesà la section Référence des AWS CLI commandes.

`batch-disable-standards`

L'exemple de code suivant montre comment utiliser`batch-disable-standards`.

AWS CLI

Pour désactiver une norme

L'`batch-disable-standards`exemple suivant désactive la norme associée à l'ARN d'abonnement spécifié.

```
aws securityhub batch-disable-standards \
    --standards-subscription-arns "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

Sortie :

```
{
    "StandardsSubscriptions": [
        {
            "StandardsArn": "arn:aws:securityhub:eu-central-1::standards/pci-dss/
v/3.2.1",
            "StandardsInput": { },
```

```
            "StandardsStatus": "DELETING",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Désactivation ou activation d'une norme de sécurité dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchDisableStandardsà la section Référence des AWS CLI commandes.

`batch-enable-standards`

L'exemple de code suivant montre comment utiliser`batch-enable-standards`.

AWS CLI

Pour activer une norme

L'`batch-enable-standards`exemple suivant active la norme PCI DSS pour le compte demandeur.

```
aws securityhub batch-enable-standards \
    --standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-
west-1::standards/pci-dss/v/3.2.1"}'
```

Sortie :

```
{
    "StandardsSubscriptions": [
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "StandardsInput": { },
            "StandardsStatus": "PENDING",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Désactivation ou activation d'une norme de sécurité dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchEnableStandardsà la section Référence des AWS CLI commandes.

`batch-get-automation-rules`

L'exemple de code suivant montre comment utiliser`batch-get-automation-rules`.

AWS CLI

Pour obtenir des informations détaillées sur les règles d'automatisation

L'`batch-get-automation-rules`exemple suivant permet d'obtenir les détails de la règle d'automatisation spécifiée. Vous pouvez obtenir des informations détaillées sur une ou plusieurs règles d'automatisation à l'aide d'une seule commande.

```
aws securityhub batch-get-automation-rules \
    --automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

Sortie :

```
{
    "Rules": [
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "Suppress informational findings",
            "Description": "Suppress GuardDuty findings with Informational
 severity",
            "IsTerminal": false,
            "Criteria": {
                "ProductName": [
                    {
                        "Value": "GuardDuty",
                        "Comparison": "EQUALS"
                    }
                ],
                "SeverityLabel": [
```

```
                    {
                        "Value": "INFORMATIONAL",
                        "Comparison": "EQUALS"
                    }
                ],
                "WorkflowStatus": [
                    {
                        "Value": "NEW",
                        "Comparison": "EQUALS"
                    }
                ],
                "RecordState": [
                    {
                        "Value": "ACTIVE",
                        "Comparison": "EQUALS"
                    }
                ]
            },
            "Actions": [
                {
                    "Type": "FINDING_FIELDS_UPDATE",
                    "FindingFieldsUpdate": {
                        "Note": {
                            "Text": "Automatically suppress GuardDuty findings with
 Informational severity",
                            "UpdatedBy": "sechub-automation"
                        },
                        "Workflow": {
                            "Status": "SUPPRESSED"
                        }
                    }
                }
            ],
            "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
            "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        }
    ],
    "UnprocessedAutomationRules": []
}
```

Pour plus d'informations, consultez la section Affichage des règles d'automatisation dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchGetAutomationRulesà la section Référence des AWS CLI commandes.

`batch-get-configuration-policy-associations`

L'exemple de code suivant montre comment utiliser`batch-get-configuration-policy-associations`.

AWS CLI

Pour obtenir les détails de l'association de configuration pour un lot de cibles

L'`batch-get-configuration-policy-associations`exemple suivant récupère les détails de l'association pour les cibles spécifiées. Vous pouvez fournir le compte IDs IDs, l'unité organisationnelle ou l'ID racine de la cible.

```
aws securityhub batch-get-configuration-policy-associations \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

Sortie :

```
{
    "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "TargetId": "ou-6hi7-8j91kl2m",
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
    "AssociationStatus": "SUCCESS",
    "AssociationStatusMessage": "Association applied successfully on this target."
}
```

Pour plus d'informations, consultez la section Visualisation des politiques de configuration de Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchGetConfigurationPolicyAssociationsà la section Référence des AWS CLI commandes.

`batch-get-security-controls`

L'exemple de code suivant montre comment utiliser`batch-get-security-controls`.

AWS CLI

Pour obtenir les détails des contrôles de sécurité

L'`batch-get-security-controls`exemple suivant fournit des détails sur les contrôles de sécurité ACM.1 et IAM.1 dans le compte courant AWS et la région. AWS

```
aws securityhub batch-get-security-controls \
    --security-control-ids '["ACM.1", "IAM.1"]'
```

Sortie :

```
{
    "SecurityControls": [
        {
            "SecurityControlId": "ACM.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/ACM.1",
            "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
            "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.1/remediation",
            "SeverityRating": "MEDIUM",
            "SecurityControlStatus": "ENABLED"
            "UpdateStatus": "READY",
            "Parameters": {
                "daysToExpiration": {
                    "ValueType": CUSTOM,
                    "Value": {
                        "Integer": 15
                    }
                }
            },
            "LastUpdateReason": "Updated control parameter"
        },
        {
            "SecurityControlId": "IAM.1",
```

```
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/IAM.1",
            "Title": "IAM policies should not allow full \"*\" administrative
 privileges",
            "Description": "This AWS control checks whether the default version of
 AWS Identity and Access Management (IAM) policies (also known as customer managed
 policies) do not have administrator access with a statement that has \"Effect\":
 \"Allow\" with \"Action\": \"*\" over \"Resource\": \"*\". It only checks for
 the Customer Managed Policies that you created, but not inline and AWS Managed
 Policies.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
IAM.1/remediation",
            "SeverityRating": "HIGH",
            "SecurityControlStatus": "ENABLED"
            "UpdateStatus": "READY",
            "Parameters": {}
        }
    ]
}
```

Pour plus d'informations, consultez la section Affichage des détails d'un contrôle dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchGetSecurityControlsà la section Référence des AWS CLI commandes.

`batch-get-standards-control-associations`

L'exemple de code suivant montre comment utiliser`batch-get-standards-control-associations`.

AWS CLI

Pour obtenir le statut d'activation d'un contrôle

L'`batch-get-standards-control-associations`exemple suivant indique si les contrôles spécifiés sont activés dans les normes spécifiées.

```
aws securityhub batch-get-standards-control-associations \
    --standards-control-association-ids '[{"SecurityControlId":
 "Config.1","StandardsArn": "arn:aws:securityhub:us-east-1:123456789012:ruleset/cis-
aws-foundations-benchmark/v/1.2.0"}, {"SecurityControlId": "IAM.6","StandardsArn":
```

```
 "arn:aws:securityhub:us-east-1:123456789012:standards/aws-foundational-security-best-practices/v/1.0.0"}]'
```

Sortie :

```
{
    "StandardsControlAssociationDetails": [
        {
            "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0",
            "SecurityControlId": "Config.1",
            "SecurityControlArn": "arn:aws:securityhub:us-east-1:068873283051:security-control/Config.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "CIS AWS Foundations 2.5"
            ],
            "UpdatedAt": "2022-10-27T16:07:12.960000+00:00",
            "StandardsControlTitle": "Ensure AWS Config is enabled",
            "StandardsControlDescription": "AWS Config is a web service that performs configuration management of supported AWS resources within your account and delivers log files to you. The recorded information includes the configuration item (AWS resource), relationships between configuration items (AWS resources), and any configuration changes between resources. It is recommended to enable AWS Config in all regions.",
            "StandardsControlArns": [
                "arn:aws:securityhub:us-east-1:068873283051:control/cis-aws-foundations-benchmark/v/1.2.0/2.5"
            ]
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-foundational-security-best-practices/v/1.0.0",
            "SecurityControlId": "IAM.6",
            "SecurityControlArn": "arn:aws:securityhub:us-east-1:068873283051:security-control/IAM.6",
            "AssociationStatus": "DISABLED",
            "RelatedRequirements": [],
            "UpdatedAt": "2022-11-22T21:30:35.080000+00:00",
            "UpdatedReason": "test",
            "StandardsControlTitle": "Hardware MFA should be enabled for the root user",
```

```
            "StandardsControlDescription": "This AWS control checks whether your AWS
 account is enabled to use a hardware multi-factor authentication (MFA) device to
 sign in with root user credentials.",
            "StandardsControlArns": [
                "arn:aws:securityhub:us-east-1:068873283051:control/aws-
foundational-security-best-practices/v/1.0.0/IAM.6"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation et désactivation des contrôles dans des normes spécifiques dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchGetStandardsControlAssociationsà la section Référence des AWS CLI commandes.

`batch-import-findings`

L'exemple de code suivant montre comment utiliser`batch-import-findings`.

AWS CLI

Pour mettre à jour un résultat

L'`batch-import-findings`exemple suivant met à jour un résultat.

```
aws securityhub batch-import-findings \
     --findings '
        [{
            "AwsAccountId": "123456789012",
            "CreatedAt": "2020-05-27T17:05:54.832Z",
            "Description": "Vulnerability in a CloudTrail trail",
            "FindingProviderFields": {
                "Severity": {
                    "Label": "LOW",
                    "Original": "10"
                },
                "Types": [
                    "Software and Configuration Checks/Vulnerabilities/CVE"
                ]
            },
            "GeneratorId": "TestGeneratorId",
```

```
            "Id": "Id1",
            "ProductArn": "arn:aws:securityhub:us-
west-1:123456789012:product/123456789012/default",
            "Resources": [
                {
                    "Id": "arn:aws:cloudtrail:us-west-1:123456789012:trail/
TrailName",
                    "Partition": "aws",
                    "Region": "us-west-1",
                    "Type": "AwsCloudTrailTrail"
                }
            ],
            "SchemaVersion": "2018-10-08",
            "Title": "CloudTrail trail vulnerability",
            "UpdatedAt": "2020-06-02T16:05:54.832Z"
        }]'
```

Sortie :

```
{
    "FailedCount": 0,
    "SuccessCount": 1,
    "FailedFindings": []
}
```

Pour plus d'informations, consultez la section Utiliser BatchImportFindings pour créer et mettre à jour des résultats dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchImportFindingsà la section Référence des AWS CLI commandes.

`batch-update-automation-rules`

L'exemple de code suivant montre comment utiliser`batch-update-automation-rules`.

AWS CLI

Pour mettre à jour les règles d'automatisation

L'`batch-update-automation-rules`exemple suivant met à jour la règle d'automatisation spécifiée. Vous pouvez mettre à jour une ou plusieurs règles à l'aide d'une seule commande. Seul le compte administrateur du Security Hub peut exécuter cette commande.

```
aws securityhub batch-update-automation-rules \
    --update-automation-rules-request-items '[ \
        { \
            "Actions": [{ \
                "Type": "FINDING_FIELDS_UPDATE", \
                "FindingFieldsUpdate": { \
                    "Note": { \
                        "Text": "Known issue that is a risk", \
                        "UpdatedBy": "sechub-automation" \
                    }, \
                    "Workflow": { \
                        "Status": "NEW" \
                    } \
                } \
            }], \
            "Criteria": { \
                "SeverityLabel": [{ \
                    "Value": "LOW", \
                    "Comparison": "EQUALS" \
                }] \
            }, \
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", \
            "RuleOrder": 1, \
            "RuleStatus": "DISABLED" \
        } \
    ]'
```

Sortie :

```
{
    "ProcessedAutomationRules": [
        "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
}
```

Pour plus d'informations, consultez la section Modification des règles d'automatisation dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchUpdateAutomationRulesà la section Référence des AWS CLI commandes.

`batch-update-findings`

L'exemple de code suivant montre comment utiliser`batch-update-findings`.

AWS CLI

Exemple 1 : pour mettre à jour un résultat

L'`batch-update-findings`exemple suivant met à jour deux résultats pour ajouter une note, modifier l'étiquette de gravité et résoudre le problème.

```
aws securityhub batch-update-findings \
    --finding-identifiers '[{"Id": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", "ProductArn": "arn:aws:securityhub:us-
west-1::product/aws/securityhub"}, {"Id": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222", "ProductArn": "arn:aws:securityhub:us-
west-1::product/aws/securityhub"}]' \
    --note '{"Text": "Known issue that is not a risk.", "UpdatedBy": "user1"}' \
    --severity '{"Label": "LOW"}' \
    --workflow '{"Status": "RESOLVED"}'
```

Sortie :

```
{
    "ProcessedFindings": [
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        },
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        }
    ],
    "UnprocessedFindings": []
}
```

Pour plus d'informations, consultez la section Utiliser BatchUpdateFindings pour mettre à jour un résultat dans le Guide de l'utilisateur du AWS Security Hub.

Exemple 2 : Pour mettre à jour un résultat à l'aide d'une syntaxe abrégée

L'`batch-update-findings`exemple suivant met à jour deux résultats pour ajouter une note, modifier l'étiquette de gravité et la résoudre à l'aide d'une syntaxe abrégée.

```
aws securityhub batch-update-findings \
    --finding-identifiers Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" \
    --note Text="Known issue that is not a risk.",UpdatedBy="user1" \
    --severity Label="LOW" \
    --workflow Status="RESOLVED"
```

Sortie :

```
{
    "ProcessedFindings": [
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        },
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        }
    ],
    "UnprocessedFindings": []
}
```

Pour plus d'informations, consultez la section Utiliser BatchUpdateFindings pour mettre à jour un résultat dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchUpdateFindingsà la section Référence des AWS CLI commandes.

`batch-update-standards-control-associations`

L'exemple de code suivant montre comment utiliser`batch-update-standards-control-associations`.

AWS CLI

Pour mettre à jour le statut d'activation d'un contrôle dans les normes activées

L'`batch-update-standards-control-associations`exemple suivant désactive la version CloudTrail .1 dans les normes spécifiées.

```
aws securityhub batch-update-standards-control-associations \
    --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",
 "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable
 to environment"}, {"SecurityControlId": "CloudTrail.1", "StandardsArn":
 "arn:aws:securityhub:::standards/cis-aws-foundations-benchmark/v/1.4.0",
 "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable to
 environment"}]'
```

Cette commande ne produit aucune sortie lorsqu'elle réussit.

Pour plus d'informations, consultez les sections Activation et désactivation des contrôles dans des normes spécifiques et Activation et désactivation des contrôles dans toutes les normes dans le Guide de l'utilisateur AWS de Security Hub.

- Pour plus de détails sur l'API, reportez-vous BatchUpdateStandardsControlAssociationsà la section Référence des AWS CLI commandes.

`create-action-target`

L'exemple de code suivant montre comment utiliser`create-action-target`.

AWS CLI

Pour créer une action personnalisée

L'`create-action-target`exemple suivant crée une action personnalisée. Il fournit le nom, la description et l'identifiant de l'action.

```
aws securityhub create-action-target \
```

```
    --name "Send to remediation" \
    --description "Action to send the finding for remediation tracking" \
    --id "Remediation"
```

Sortie :

```
{
    "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
}
```

Pour plus d'informations, consultez la section Création d'une action personnalisée et association de celle-ci à une règle d' CloudWatch événements dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous CreateActionTargetà la section Référence des AWS CLI commandes.

`create-automation-rule`

L'exemple de code suivant montre comment utiliser`create-automation-rule`.

AWS CLI

Pour créer une règle d'automatisation

L'`create-automation-rule`exemple suivant crée une règle d'automatisation dans le AWS compte courant et dans AWS la région. Security Hub filtre vos résultats en fonction des critères spécifiés et applique les actions aux résultats correspondants. Seul le compte administrateur du Security Hub peut exécuter cette commande.

```
aws securityhub create-automation-rule \
    --actions '[{ \
        "Type": "FINDING_FIELDS_UPDATE", \
        "FindingFieldsUpdate": { \
            "Severity": { \
                "Label": "HIGH" \
            }, \
            "Note": { \
                "Text": "Known issue that is a risk. Updated by automation rules", \
                "UpdatedBy": "sechub-automation" \
```

```
            } \
        } \
    }]' \
    --criteria '{ \
        "SeverityLabel": [{ \
            "Value": "INFORMATIONAL", \
            "Comparison": "EQUALS" \
        }] \
    }' \
    --description "A sample rule" \
    --no-is-terminal \
    --rule-name "sample rule" \
    --rule-order 1 \
    --rule-status "ENABLED"
```

Sortie :

```
{
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Création de règles d'automatisation dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous CreateAutomationRuleà la section Référence des AWS CLI commandes.

`create-configuration-policy`

L'exemple de code suivant montre comment utiliser`create-configuration-policy`.

AWS CLI

Pour créer une politique de configuration

L'`create-configuration-policy`exemple suivant crée une politique de configuration avec les paramètres spécifiés.

```
aws securityhub create-configuration-policy \
    --name "SampleConfigurationPolicy" \
    --description "SampleDescription" \
```

```
    --configuration-policy '{"SecurityHub": {"ServiceEnabled":
 true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-
central-1::standards/aws-foundational-security-best-practices/
v/1.0.0","arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"],"SecurityControlsConfiguration":{"DisabledSecurityControlIdentifiers":
 ["CloudTrail.2"], "SecurityControlCustomParameters": [{"SecurityControlId":
 "ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value":
 {"Integer": 15}}}}]}}}' \
    --tags '{"Environment": "Prod"}'
```

Sortie :

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "SampleConfigurationPolicy",
    "Description": "SampleDescription",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
                            }
```

```
                        }
                    }
                ]
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Création et association de politiques de configuration de Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous CreateConfigurationPolicyà la section Référence des AWS CLI commandes.

`create-finding-aggregator`

L'exemple de code suivant montre comment utiliser`create-finding-aggregator`.

AWS CLI

Pour activer l'agrégation de recherche

L'`create-finding-aggregator`exemple suivant configure la recherche d'une agrégation. Il est géré depuis l'est des États-Unis (Virginie), qui désigne l'est des États-Unis (Virginie) comme région d'agrégation. Cela indique de ne lier que les régions spécifiées et de ne pas lier automatiquement de nouvelles régions. Elle sélectionne l'ouest des États-Unis (Californie du Nord) et l'ouest des États-Unis (Oregon) comme régions liées.

```
aws securityhub create-finding-aggregator \
    --region us-east-1 \
    --region-linking-mode SPECIFIED_REGIONS \
    --regions us-west-1,us-west-2
```

Sortie :

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000",
    "FindingAggregationRegion": "us-east-1",
    "RegionLinkingMode": "SPECIFIED_REGIONS",
    "Regions": "us-west-1,us-west-2"
```

```
}
```

Pour plus d'informations, consultez la section Activation de l'agrégation des recherches dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous CreateFindingAggregatorà la section Référence des AWS CLI commandes.

`create-insight`

L'exemple de code suivant montre comment utiliser`create-insight`.

AWS CLI

Pour créer un aperçu personnalisé

L'`create-insight`exemple suivant crée un aperçu personnalisé nommé Critical role findings qui renvoie des résultats critiques liés aux AWS rôles.

```
aws securityhub create-insight \
    --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \
    --group-by-attribute "ResourceId" \
    --name "Critical role findings"
```

Sortie :

```
{
    "InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Gestion des informations personnalisées dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous CreateInsightà la section Référence des AWS CLI commandes.

`create-members`

L'exemple de code suivant montre comment utiliser`create-members`.

AWS CLI

Pour ajouter des comptes en tant que comptes de membre

L'`create-members`exemple suivant ajoute deux comptes en tant que comptes membres au compte administrateur demandeur.

```
aws securityhub create-members \
    --account-details '[{"AccountId": "123456789111"}, {"AccountId":
 "123456789222"}]'
```

Sortie :

```
{
    "UnprocessedAccounts": []
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous CreateMembersà la section Référence des AWS CLI commandes.

`decline-invitations`

L'exemple de code suivant montre comment utiliser`decline-invitations`.

AWS CLI

Pour refuser une invitation à devenir membre

L'`decline-invitations`exemple suivant refuse une invitation à devenir membre du compte administrateur spécifié. Le compte membre est le compte demandeur.

```
aws securityhub decline-invitations \
    --account-ids "123456789012"
```

Sortie :

```
{
    "UnprocessedAccounts": []
```

```
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeclineInvitationsà la section Référence des AWS CLI commandes.

`delete-action-target`

L'exemple de code suivant montre comment utiliser`delete-action-target`.

AWS CLI

Pour supprimer une action personnalisée

L'`delete-action-target`exemple suivant supprime l'action personnalisée identifiée par l'ARN spécifié.

```
aws securityhub delete-action-target \
    --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
```

Sortie :

```
{
    "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
}
```

Pour plus d'informations, consultez la section Création d'une action personnalisée et association de celle-ci à une règle d' CloudWatch événements dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeleteActionTargetà la section Référence des AWS CLI commandes.

`delete-configuration-policy`

L'exemple de code suivant montre comment utiliser`delete-configuration-policy`.

AWS CLI

Pour supprimer une politique de configuration

L'`delete-configuration-policy`exemple suivant supprime la politique de configuration spécifiée.

```
aws securityhub delete-configuration-policy \
    --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression et dissociation des politiques de configuration de Security Hub dans le Guide de l'utilisateur AWS de Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeleteConfigurationPolicyà la section Référence des AWS CLI commandes.

`delete-finding-aggregator`

L'exemple de code suivant montre comment utiliser`delete-finding-aggregator`.

AWS CLI

Pour arrêter de trouver une agrégation

L'`delete-finding-aggregator`exemple suivant arrête de rechercher l'agrégation. Il est géré depuis l'est des États-Unis (Virginie), qui est la région d'agrégation.

```
aws securityhub delete-finding-aggregator \
    --region us-east-1 \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Arrêter de rechercher l'agrégation dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeleteFindingAggregatorà la section Référence des AWS CLI commandes.

`delete-insight`

L'exemple de code suivant montre comment utiliser`delete-insight`.

AWS CLI

Pour supprimer un aperçu personnalisé

L'`delete-insight`exemple suivant supprime l'aperçu personnalisé avec l'ARN spécifié.

```
aws securityhub delete-insight \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Sortie :

```
{
   "InsightArn": "arn:aws:securityhub:eu-
central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

Pour plus d'informations, consultez la section Gestion des informations personnalisées dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeleteInsightà la section Référence des AWS CLI commandes.

`delete-invitations`

L'exemple de code suivant montre comment utiliser`delete-invitations`.

AWS CLI

Pour supprimer une invitation à devenir membre

L'`delete-invitations`exemple suivant supprime une invitation à devenir membre pour le compte administrateur spécifié. Le compte membre est le compte demandeur.

```
aws securityhub delete-invitations \
    --account-ids "123456789012"
```

Sortie :

```
{
    "UnprocessedAccounts": []
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeleteInvitationsà la section Référence des AWS CLI commandes.

`delete-members`

L'exemple de code suivant montre comment utiliser`delete-members`.

AWS CLI

Pour supprimer des comptes de membres

L'`delete-members`exemple suivant supprime les comptes de membres spécifiés du compte d'administrateur demandeur.

```
aws securityhub delete-members \
    --account-ids "123456789111" "123456789222"
```

Sortie :

```
{
    "UnprocessedAccounts": []
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DeleteMembersà la section Référence des AWS CLI commandes.

`describe-action-targets`

L'exemple de code suivant montre comment utiliser`describe-action-targets`.

AWS CLI

Pour récupérer des informations sur les actions personnalisées

L'`describe-action-targets`exemple suivant récupère des informations sur l'action personnalisée identifiée par l'ARN spécifié.

```
aws securityhub describe-action-targets \
    --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
```

Sortie :

```
{
    "ActionTargets": [
        {
            "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/
custom/Remediation",
            "Description": "Action to send the finding for remediation tracking",
            "Name": "Send to remediation"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création d'une action personnalisée et association de celle-ci à une règle d' CloudWatch événements dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DescribeActionTargetsà la section Référence des AWS CLI commandes.

`describe-hub`

L'exemple de code suivant montre comment utiliser`describe-hub`.

AWS CLI

Pour obtenir des informations sur une ressource du hub

L'`describe-hub`exemple suivant renvoie la date d'abonnement pour la ressource hub spécifiée. La ressource du hub est identifiée par son ARN.

```
aws securityhub describe-hub \
    --hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Sortie :

```
{
    "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",
    "SubscribedAt": "2019-11-19T23:15:10.046Z"
}
```

Pour plus d'informations, consultez AWS: : SecurityHub : :Hub dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous DescribeHubà la section Référence des AWS CLI commandes.

`describe-organization-configuration`

L'exemple de code suivant montre comment utiliser`describe-organization-configuration`.

AWS CLI

Pour voir comment Security Hub est configuré pour une organisation

L'`describe-organization-configuration`exemple suivant renvoie des informations sur la manière dont une organisation est configurée dans Security Hub. Dans cet exemple, l'organisation utilise une configuration centralisée. Seul le compte administrateur du Security Hub peut exécuter cette commande.

```
aws securityhub describe-organization-configuration
```

Sortie :

```
{
    "AutoEnable": false,
    "MemberAccountLimitReached": false,
    "AutoEnableStandards": "NONE",
    "OrganizationConfiguration": {
        "ConfigurationType": "LOCAL",
        "Status": "ENABLED",
```

```
        "StatusMessage": "Central configuration has been enabled successfully"
    }
}
```

Pour plus d'informations, consultez la section Gestion des comptes auprès d' AWS Organizations dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DescribeOrganizationConfigurationà la section Référence des AWS CLI commandes.

`describe-products`

L'exemple de code suivant montre comment utiliser`describe-products`.

AWS CLI

Pour renvoyer des informations sur les intégrations de produits disponibles

L'`describe-products`exemple suivant renvoie les intégrations de produits disponibles une par une.

```
aws securityhub describe-products \
    --max-results 1
```

Sortie :

```
{
    "NextToken": "U2FsdGVkX18vvPlOqb7RDrWRWVFBJI46MOIAb+nZmRJmR15NoRi2gm13sdQEn3O/
pq/78dGs+bKpgA+7HMPHO0qX33/zoRI+uIG/F9yLNhcOrOWzFUdy36JcXLQji3Rpnn/
cD1SVkGA98qI3zPOSDg==",
    "Products": [
        {
            "ProductArn": "arn:aws:securityhub:us-west-1:123456789333:product/
crowdstrike/crowdstrike-falcon",
            "ProductName": "CrowdStrike Falcon",
            "CompanyName": "CrowdStrike",
            "Description": "CrowdStrike Falcon's single lightweight sensor unifies
 next-gen antivirus, endpoint detection and response, and 24/7 managed hunting, via
 the cloud.",
            "Categories": [
                "Endpoint Detection and Response (EDR)",
                "AV Scanning and Sandboxing",
```

```
                "Threat Intelligence Feeds and Reports",
                "Endpoint Forensics",
                "Network Forensics"
            ],
            "IntegrationTypes": [
                "SEND_FINDINGS_TO_SECURITY_HUB"
            ],
            "MarketplaceUrl": "https://aws.amazon.com/marketplace/seller-profile?
id=a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ActivationUrl": "https://falcon.crowdstrike.com/support/documentation",
            "ProductSubscriptionResourcePolicy": "{\"Version\":
\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"123456789333\"},\"Action\":[\"securityhub:BatchImportFindings\"],\"Resource\":
\"arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/
crowdstrike-falcon\",\"Condition\":{\"StringEquals\":{\"securityhub:TargetAccount
\":\"123456789012\"}}},{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"123456789012\"},\"Action\":[\"securityhub:BatchImportFindings\"],\"Resource
\":\"arn:aws:securityhub:us-west-1:123456789333:product/crowdstrike/crowdstrike-
falcon\",\"Condition\":{\"StringEquals\":{\"securityhub:TargetAccount\":
\"123456789012\"}}}]}"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des intégrations de produits dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DescribeProductsà la section Référence des AWS CLI commandes.

`describe-standards-controls`

L'exemple de code suivant montre comment utiliser`describe-standards-controls`.

AWS CLI

Pour demander la liste des contrôles dans une norme activée

L'`describe-standards-controls`exemple suivant demande la liste des contrôles figurant dans l'abonnement du compte demandeur à la norme PCI DSS. La demande renvoie deux contrôles à la fois.

```
aws securityhub describe-standards-controls \
```

```
    --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1" \
    --max-results 2
```

Sortie :

```
{
    "Controls": [
        {
            "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1",
            "ControlStatus": "ENABLED",
            "ControlStatusUpdatedAt": "2020-05-15T18:49:04.473000+00:00",
            "ControlId": "PCI.AutoScaling.1",
            "Title": "Auto scaling groups associated with a load balancer should use
 health checks",
            "Description": "This AWS control checks whether your Auto Scaling groups
 that are associated with a load balancer are using Elastic Load Balancing health
 checks.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.AutoScaling.1/remediation",
            "SeverityRating": "LOW",
            "RelatedRequirements": [
                "PCI DSS 2.2"
            ]
        },
        {
            "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.CW.1",
            "ControlStatus": "ENABLED",
            "ControlStatusUpdatedAt": "2020-05-15T18:49:04.498000+00:00",
            "ControlId": "PCI.CW.1",
            "Title": "A log metric filter and alarm should exist for usage of the
 \"root\" user",
            "Description": "This control checks for the CloudWatch metric
 filters using the following pattern { $.userIdentity.type = \"Root\" &&
 $.userIdentity.invokedBy NOT EXISTS && $.eventType != \"AwsServiceEvent\" }
 It checks that the log group name is configured for use with active multi-
region CloudTrail, that there is at least one Event Selector for a Trail with
 IncludeManagementEvents set to true and ReadWriteType set to All, and that there is
 at least one active subscriber to an SNS topic associated with the alarm.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.CW.1/remediation",
```

```
            "SeverityRating": "MEDIUM",
            "RelatedRequirements": [
                "PCI DSS 7.2.1"
            ]
        }
    ],
    "NextToken": "U2FsdGVkX1+eNkPoZHVl11ip5HUYQPWSWZGmftcmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveoOdvfxXCkOBaGhohIXhsIedN+LSjQV/
l7kfCfJcq4PziNC1N9xe9aq2pjlLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUVoeriHZhyIiBgWKf7oNYdBVG8OEortVWvSkoUTt
+B2ThcnC7l43kI0UNxlkZ6sc64AsW"
}
```

Pour plus d'informations, consultez la section Affichage des informations relatives aux contrôles dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DescribeStandardsControlsà la section Référence des AWS CLI commandes.

`describe-standards`

L'exemple de code suivant montre comment utiliser`describe-standards`.

AWS CLI

Pour renvoyer une liste des normes disponibles

L'`describe-standards`exemple suivant renvoie la liste des normes disponibles.

```
aws securityhub describe-standards
```

Sortie :

```
{
    "Standards": [
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/aws-
foundational-security-best-practices/v/1.0.0",
            "Name": "AWS Foundational Security Best Practices v1.0.0",
            "Description": "The AWS Foundational Security Best Practices standard
 is a set of automated security checks that detect when AWS accounts and deployed
 resources do not align to security best practices. The standard is defined by AWS
```

```
 security experts. This curated set of controls helps improve your security posture
 in AWS, and cover AWS's most popular and foundational services.",
            "EnabledByDefault": true
        },
        {
            "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
            "Name": "CIS AWS Foundations Benchmark v1.2.0",
            "Description": "The Center for Internet Security (CIS) AWS Foundations
 Benchmark v1.2.0 is a set of security configuration best practices for AWS. This
 Security Hub standard automatically checks for your compliance readiness against a
 subset of CIS requirements.",
            "EnabledByDefault": true
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "Name": "PCI DSS v3.2.1",
            "Description": "The Payment Card Industry Data Security Standard (PCI
 DSS) v3.2.1 is an information security standard for entities that store, process,
 and/or transmit cardholder data. This Security Hub standard automatically checks
 for your compliance readiness against a subset of PCI DSS requirements.",
            "EnabledByDefault": false
        }
    ]
}
```

Pour plus d'informations, consultez la section Normes de sécurité AWS de Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DescribeStandardsà la section Référence des AWS CLI commandes.

`disable-import-findings-for-product`

L'exemple de code suivant montre comment utiliser`disable-import-findings-for-product`.

AWS CLI

Pour ne plus recevoir les résultats de l'intégration d'un produit

L'`disable-import-findings-for-product`exemple suivant désactive le flux de résultats pour l'abonnement spécifié à une intégration de produit.

```
aws securityhub disable-import-findings-for-product \
    --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des intégrations de produits dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DisableImportFindingsForProductà la section Référence des AWS CLI commandes.

`disable-organization-admin-account`

L'exemple de code suivant montre comment utiliser`disable-organization-admin-account`.

AWS CLI

Pour supprimer un compte administrateur Security Hub

L'`disable-organization-admin-account`exemple suivant révoque l'attribution du compte spécifié en tant que compte administrateur Security Hub pour AWS Organizations.

```
aws securityhub disable-organization-admin-account \
    --admin-account-id 777788889999
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désignation d'un compte administrateur Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DisableOrganizationAdminAccountà la section Référence des AWS CLI commandes.

`disable-security-hub`

L'exemple de code suivant montre comment utiliser`disable-security-hub`.

AWS CLI

Pour désactiver AWS Security Hub

L'`disable-security-hub`exemple suivant désactive AWS Security Hub pour le compte demandeur.

```
aws securityhub disable-security-hub
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation AWS de Security Hub dans le Guide de l'utilisateur AWS de Security Hub.

- Pour plus de détails sur l'API, reportez-vous DisableSecurityHubà la section Référence des AWS CLI commandes.

`disassociate-from-administrator-account`

L'exemple de code suivant montre comment utiliser`disassociate-from-administrator-account`.

AWS CLI

Pour se dissocier d'un compte administrateur

L'`disassociate-from-administrator-account`exemple suivant dissocie le compte demandeur de son compte administrateur actuel.

```
aws securityhub disassociate-from-administrator-account
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DisassociateFromAdministratorAccountà la section Référence des AWS CLI commandes.

`disassociate-from-master-account`

L'exemple de code suivant montre comment utiliser`disassociate-from-master-account`.

AWS CLI

Pour se dissocier d'un compte administrateur

L'`disassociate-from-master-account`exemple suivant dissocie le compte demandeur de son compte administrateur actuel.

```
aws securityhub disassociate-from-master-account
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DisassociateFromMasterAccountà la section Référence des AWS CLI commandes.

`disassociate-members`

L'exemple de code suivant montre comment utiliser`disassociate-members`.

AWS CLI

Pour dissocier les comptes des membres

L'`disassociate-members`exemple suivant dissocie les comptes de membre spécifiés du compte d'administrateur demandeur.

```
aws securityhub disassociate-members  \
    --account-ids "123456789111" "123456789222"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous DisassociateMembersà la section Référence des AWS CLI commandes.

`enable-import-findings-for-product`

L'exemple de code suivant montre comment utiliser`enable-import-findings-for-product`.

AWS CLI

Pour commencer à recevoir les résultats de l'intégration d'un produit

L'`enable-import-findings-for-product`exemple suivant active le flux des résultats issus de l'intégration de produits spécifiée.

```
aws securityhub enable-import-findings-for-product \
    --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/crowdstrike-falcon"
```

Sortie :

```
{
    "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon"
}
```

Pour plus d'informations, consultez la section Gestion des intégrations de produits dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous EnableImportFindingsForProductà la section Référence des AWS CLI commandes.

`enable-organization-admin-account`

L'exemple de code suivant montre comment utiliser`enable-organization-admin-account`.

AWS CLI

Pour désigner un compte d'organisation en tant que compte administrateur du Security Hub

L'`enable-organization-admin-account`exemple suivant désigne le compte spécifié en tant que compte administrateur du Security Hub.

```
aws securityhub enable-organization-admin-account \
    --admin-account-id 777788889999
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désignation d'un compte administrateur Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous EnableOrganizationAdminAccountà la section Référence des AWS CLI commandes.

`enable-security-hub`

L'exemple de code suivant montre comment utiliser`enable-security-hub`.

AWS CLI

Pour activer AWS Security Hub

L'`enable-security-hub`exemple suivant active AWS Security Hub pour le compte demandeur. Il configure Security Hub pour activer les normes par défaut. Pour la ressource du hub, il attribue la valeur `Security` à la balise`Department`.

```
aws securityhub enable-security-hub \
    --enable-default-standards \
    --tags '{"Department": "Security"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Enabling Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous EnableSecurityHubà la section Référence des AWS CLI commandes.

`get-administrator-account`

L'exemple de code suivant montre comment utiliser`get-administrator-account`.

AWS CLI

Pour récupérer les informations relatives à un compte administrateur

L'`get-administrator-account`exemple suivant permet de récupérer des informations sur le compte administrateur du compte demandeur.

```
aws securityhub get-administrator-account
```

Sortie :

```
{
   "Master": {
      "AccountId": "123456789012",
```

```
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
   }
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetAdministratorAccountà la section Référence des AWS CLI commandes.

`get-configuration-policy-association`

L'exemple de code suivant montre comment utiliser`get-configuration-policy-association`.

AWS CLI

Pour obtenir les détails de l'association de configuration pour une cible

L'`get-configuration-policy-association`exemple suivant récupère les détails de l'association pour la cible spécifiée. Vous pouvez fournir un identifiant de compte, un identifiant d'unité organisationnelle ou l'identifiant racine de la cible.

```
aws securityhub get-configuration-policy-association \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

Sortie :

```
{
    "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "TargetId": "ou-6hi7-8j91kl2m",
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
    "AssociationStatus": "SUCCESS",
    "AssociationStatusMessage": "Association applied successfully on this target."
}
```

Pour plus d'informations, consultez la section Visualisation des politiques de configuration de Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetConfigurationPolicyAssociationà la section Référence des AWS CLI commandes.

`get-configuration-policy`

L'exemple de code suivant montre comment utiliser`get-configuration-policy`.

AWS CLI

Pour afficher les détails de la politique de configuration

L'`get-configuration-policy`exemple suivant permet de récupérer les détails de la politique de configuration spécifiée.

```
aws securityhub get-configuration-policy \
   --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Sortie :

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "ce5ed1e7-9639-4e2f-9313-fa87fcef944b",
    "Name": "SampleConfigurationPolicy",
    "Description": "SampleDescription",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
```

```
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Visualisation des politiques de configuration de Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetConfigurationPolicyà la section Référence des AWS CLI commandes.

`get-enabled-standards`

L'exemple de code suivant montre comment utiliser`get-enabled-standards`.

AWS CLI

Pour récupérer des informations sur une norme activée

L'`get-enabled-standards`exemple suivant permet de récupérer des informations sur la norme PCI DSS.

```
aws securityhub get-enabled-standards \
    --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

Sortie :

```
{
    "StandardsSubscriptions": [
```

```
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "StandardsInput": { },
            "StandardsStatus": "READY",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Normes de sécurité AWS de Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetEnabledStandardsà la section Référence des AWS CLI commandes.

`get-finding-aggregator`

L'exemple de code suivant montre comment utiliser`get-finding-aggregator`.

AWS CLI

Pour récupérer la configuration actuelle de l'agrégation des résultats

L'`get-finding-aggregator`exemple suivant récupère la configuration actuelle de l'agrégation des résultats.

```
aws securityhub get-finding-aggregator \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000
```

Sortie :

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000",
    "FindingAggregationRegion": "us-east-1",
    "RegionLinkingMode": "SPECIFIED_REGIONS",
    "Regions": "us-west-1,us-west-2"
}
```

Pour plus d'informations, consultez la section Affichage de la configuration actuelle de l'agrégation des résultats dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetFindingAggregatorà la section Référence des AWS CLI commandes.

`get-finding-history`

L'exemple de code suivant montre comment utiliser`get-finding-history`.

AWS CLI

Pour accéder à l'historique des recherches

L'`get-finding-history`exemple suivant permet d'obtenir l'historique des 90 derniers jours pour le résultat spécifié. Dans cet exemple, les résultats sont limités à deux enregistrements de l'historique des recherches.

```
aws securityhub get-finding-history \
    --finding-identifier Id="arn:aws:securityhub:us-
east-1:123456789012:security-control/S3.17/finding/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",ProductArn="arn:aws:securityhub:us-east-1::product/aws/securityhub"
```

Sortie :

```
{
    "Records": [
        {
            "FindingIdentifier": {
                "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
securityhub"
            },
            "UpdateTime": "2023-06-02T03:15:25.685000+00:00",
            "FindingCreated": false,
            "UpdateSource": {
                "Type": "BATCH_IMPORT_FINDINGS",
                "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
            },
            "Updates": [
                {
                    "UpdatedField": "Compliance.RelatedRequirements",
```

```
                    "OldValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
 SC-12(3)\",\"NIST.800-53.r5 SC-12(6)\",\"NIST.800-53.r5 CM-3(6)\",\"NIST.800-53.r5
 SC-13\",\"NIST.800-53.r5 SC-28\",\"NIST.800-53.r5 SC-28(1)\",\"NIST.800-53.r5
 SC-7(10)\"]",
                    "NewValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
 CM-3(6)\",\"NIST.800-53.r5 SC-13\",\"NIST.800-53.r5 SC-28\",\"NIST.800-53.r5
 SC-28(1)\",\"NIST.800-53.r5 SC-7(10)\",\"NIST.800-53.r5 CA-9(1)\",\"NIST.800-53.r5
 SI-7(6)\",\"NIST.800-53.r5 AU-9\"]"
                },
                {
                    "UpdatedField": "LastObservedAt",
                    "OldValue": "2023-06-01T09:15:38.587Z",
                    "NewValue": "2023-06-02T03:15:22.946Z"
                },
                {
                    "UpdatedField": "UpdatedAt",
                    "OldValue": "2023-06-01T09:15:31.049Z",
                    "NewValue": "2023-06-02T03:15:14.861Z"
                },
                {
                    "UpdatedField": "ProcessedAt",
                    "OldValue": "2023-06-01T09:15:41.058Z",
                    "NewValue": "2023-06-02T03:15:25.685Z"
                }
            ]
        },
        {
            "FindingIdentifier": {
                "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
securityhub"
            },
            "UpdateTime": "2023-05-23T02:06:51.518000+00:00",
            "FindingCreated": "true",
            "UpdateSource": {
                "Type": "BATCH_IMPORT_FINDINGS",
                "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
            },
            "Updates": []
        }
    ]
}
```

Pour plus d'informations, consultez la section Recherche de l'historique dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetFindingHistoryà la section Référence des AWS CLI commandes.

`get-findings`

L'exemple de code suivant montre comment utiliser`get-findings`.

AWS CLI

Exemple 1 : Pour renvoyer les résultats générés pour une norme spécifique

L'`get-findings`exemple suivant renvoie les résultats relatifs à la norme PCI DSS.

```
aws securityhub get-findings \
    --filters '{"GeneratorId":[{"Value": "pci-dss","Comparison":"PREFIX"}]}' \
    --max-items 1
```

Sortie :

```
{
    "Findings": [
        {
            "SchemaVersion": "2018-10-08",
            "Id": "arn:aws:securityhub:eu-central-1:123456789012:subscription/pci-
dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub",
            "GeneratorId": "pci-dss/v/3.2.1/PCI.Lambda.2",
            "AwsAccountId": "123456789012",
            "Types": [
                "Software and Configuration Checks/Industry and Regulatory
 Standards/PCI-DSS"
            ],
            "FindingProviderFields": {
                "Severity": {
                    "Original": 0,
                    "Label": "INFORMATIONAL"
                },
                "Types": [
                    "Software and Configuration Checks/Industry and Regulatory
 Standards/PCI-DSS"
```

```
                ]
            },
            "FirstObservedAt": "2020-06-02T14:02:49.159Z",
            "LastObservedAt": "2020-06-02T14:02:52.397Z",
            "CreatedAt": "2020-06-02T14:02:49.159Z",
            "UpdatedAt": "2020-06-02T14:02:52.397Z",
            "Severity": {
                "Original": 0,
                "Label": "INFORMATIONAL",
                "Normalized": 0
            },
            "Title": "PCI.Lambda.2 Lambda functions should be in a VPC",
            "Description": "This AWS control checks whether a Lambda function is in
 a VPC.",
            "Remediation": {
                "Recommendation": {
                    "Text": "For directions on how to fix this issue, please consult
 the AWS Security Hub PCI DSS documentation.",
                    "Url": "https://docs.aws.amazon.com/console/securityhub/
PCI.Lambda.2/remediation"
                }
            },
            "ProductFields": {
                "StandardsArn": "arn:aws:securityhub:::standards/pci-dss/v/3.2.1",
                "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1",
                "ControlId": "PCI.Lambda.2",
                "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/PCI.Lambda.2/remediation",
                "RelatedAWSResources:0/name": "securityhub-lambda-inside-
vpc-0e904a3b",
                "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
                "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.Lambda.2",
                "aws/securityhub/SeverityLabel": "INFORMATIONAL",
                "aws/securityhub/ProductName": "Security Hub",
                "aws/securityhub/CompanyName": "AWS",
                "aws/securityhub/FindingId": "arn:aws:securityhub:eu-
central-1::product/aws/securityhub/arn:aws:securityhub:eu-
central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
            "Resources": [
                {
```

```
                    "Type": "AwsAccount",
                    "Id": "AWS::::Account:123456789012",
                    "Partition": "aws",
                    "Region": "us-west-1"
                }
            ],
            "Compliance": {
                "Status": "PASSED",
                "RelatedRequirements": [
                    "PCI DSS 1.2.1",
                    "PCI DSS 1.3.1",
                    "PCI DSS 1.3.2",
                    "PCI DSS 1.3.4"
                ]
            },
            "WorkflowState": "NEW",
            "Workflow": {
                "Status": "NEW"
            },
            "RecordState": "ARCHIVED"
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}
```

Exemple 2 : Pour renvoyer des résultats de gravité critique dont le statut de flux de travail est NOTIFIÉ

L'`get-findings`exemple suivant renvoie des résultats dont la valeur d'étiquette de gravité est CRITICAL et le statut du flux de travail est NOTIFIÉ. Les résultats sont triés par ordre décroissant selon la valeur de Confidence.

```
aws securityhub get-findings \
    --filters '{"SeverityLabel":[{"Value":
 "CRITICAL","Comparison":"EQUALS"}],"WorkflowStatus":
 [{"Value":"NOTIFIED","Comparison":"EQUALS"}]}' \
    --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
    --max-items 1
```

Sortie :

```
{
```

```
    "Findings": [
        {
            "SchemaVersion": "2018-10-08",
            "Id": "arn:aws:securityhub:us-west-1: 123456789012:subscription/cis-aws-
foundations-benchmark/v/1.2.0/1.13/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/securityhub",
            "GeneratorId": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0/rule/1.13",
            "AwsAccountId": "123456789012",
            "Types": [
                "Software and Configuration Checks/Industry and Regulatory
 Standards/CIS AWS Foundations Benchmark"
            ],
            "FindingProviderFields" {
                "Severity": {
                    "Original": 90,
                    "Label": "CRITICAL"
                },
                "Types": [
                    "Software and Configuration Checks/Industry and Regulatory
 Standards/CIS AWS Foundations Benchmark"
                ]
            },
            "FirstObservedAt": "2020-05-21T20:16:34.752Z",
            "LastObservedAt": "2020-06-09T08:16:37.171Z",
            "CreatedAt": "2020-05-21T20:16:34.752Z",
            "UpdatedAt": "2020-06-09T08:16:36.430Z",
            "Severity": {
                "Original": 90,
                "Label": "CRITICAL",
                "Normalized": 90
            },
            "Title": "1.13 Ensure MFA is enabled for the \"root\" account",
            "Description": "The root account is the most privileged user in an AWS
 account. MFA adds an extra layer of protection on top of a user name and password.
 With MFA enabled, when a user signs in to an AWS website, they will be prompted for
 their user name and password as well as for an authentication code from their AWS
 MFA device.",
            "Remediation": {
                "Recommendation": {
                    "Text": "For directions on how to fix this issue, please consult
 the AWS Security Hub CIS documentation.",
                    "Url": "https://docs.aws.amazon.com/console/securityhub/
standards-cis-1.13/remediation"
```

```
                }
            },
            "ProductFields": {
                "StandardsGuideArn": "arn:aws:securityhub:::ruleset/cis-aws-
foundations-benchmark/v/1.2.0",
                "StandardsGuideSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0",
                "RuleId": "1.13",
                "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/standards-cis-1.13/remediation",
                "RelatedAWSResources:0/name": "securityhub-root-account-mfa-
enabled-5pftha",
                "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
                "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/cis-aws-foundations-benchmark/v/1.2.0/1.13",
                "aws/securityhub/SeverityLabel": "CRITICAL",
                "aws/securityhub/ProductName": "Security Hub",
                "aws/securityhub/CompanyName": "AWS",
                "aws/securityhub/FindingId": "arn:aws:securityhub:us-
west-1::product/aws/securityhub/arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0/1.13/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            "Resources": [
                {
                    "Type": "AwsAccount",
                    "Id": "AWS::::Account:123456789012",
                    "Partition": "aws",
                    "Region": "us-west-1"
                }
            ],
            "Compliance": {
                "Status": "FAILED"
            },
            "WorkflowState": "NEW",
            "Workflow": {
                "Status": "NOTIFIED"
            },
            "RecordState": "ACTIVE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Résultats relatifs au filtrage et au regroupement dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetFindingsà la section Référence des AWS CLI commandes.

`get-insight-results`

L'exemple de code suivant montre comment utiliser`get-insight-results`.

AWS CLI

Pour récupérer les résultats afin d'obtenir un aperçu

L'`get-insight-results`exemple suivant renvoie la liste des résultats d'analyse pour l'aperçu avec l'ARN spécifié.

```
aws securityhub get-insight-results \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Sortie :

```
{
    "InsightResults": {
        "GroupByAttribute": "ResourceId",
        "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
        "ResultValues": [
            {
                "Count": 10,
                "GroupByAttributeValue": "AWS::::Account:123456789111"
            },
            {
                "Count": 3,
                "GroupByAttributeValue": "AWS::::Account:123456789222"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Afficher les résultats et les conclusions des analyses et prendre les mesures nécessaires dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetInsightResultsà la section Référence des AWS CLI commandes.

`get-insights`

L'exemple de code suivant montre comment utiliser`get-insights`.

AWS CLI

Pour récupérer les détails d'un aperçu

L'`get-insights`exemple suivant récupère les détails de configuration de l'aperçu avec l'ARN spécifié.

```
aws securityhub get-insights \
    --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Sortie :

```
{
    "Insights": [
        {
            "Filters": {
               "ResourceType": [
                    {
                        "Comparison": "EQUALS",
                        "Value": "AwsIamRole"
                    }
                ],
                "SeverityLabel": [
                    {
                        "Comparison": "EQUALS",
                        "Value": "CRITICAL"
                    }
                ],
            },
            "GroupByAttribute": "ResourceId",
```

```
            "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
            "Name": "Critical role findings"
        }
    ]
}
```

Pour plus d'informations, consultez Insights in AWS Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetInsightsà la section Référence des AWS CLI commandes.

`get-invitations-count`

L'exemple de code suivant montre comment utiliser`get-invitations-count`.

AWS CLI

Pour récupérer le nombre d'invitations qui n'ont pas été acceptées

L'`get-invitations-count`exemple suivant permet de récupérer le nombre d'invitations que le compte demandeur a refusées ou auxquelles il n'a pas répondu.

```
aws securityhub get-invitations-count
```

Sortie :

```
{
  "InvitationsCount": 3
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetInvitationsCountà la section Référence des AWS CLI commandes.

`get-master-account`

L'exemple de code suivant montre comment utiliser`get-master-account`.

AWS CLI

Pour récupérer les informations relatives à un compte administrateur

L'`get-master-account`exemple suivant permet de récupérer des informations sur le compte administrateur du compte demandeur.

```
aws securityhub get-master-account
```

Sortie :

```
{
   "Master": {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
   }
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetMasterAccountà la section Référence des AWS CLI commandes.

`get-members`

L'exemple de code suivant montre comment utiliser`get-members`.

AWS CLI

Pour récupérer des informations sur les comptes de membres sélectionnés

L'`get-members`exemple suivant permet de récupérer des informations sur les comptes de membres spécifiés.

```
aws securityhub get-members \
    --account-ids "444455556666" "777788889999"
```

Sortie :

```
{
    "Members": [
        {
            "AccountId": "123456789111",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        },
        {
            "AccountId": "123456789222",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        }
    ],
    "UnprocessedAccounts": [ ]
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetMembersà la section Référence des AWS CLI commandes.

`get-security-control-definition`

L'exemple de code suivant montre comment utiliser`get-security-control-definition`.

AWS CLI

Pour obtenir des informations détaillées sur la définition des contrôles de sécurité

L'`get-security-control-definition`exemple suivant permet de récupérer les détails de définition d'un contrôle de sécurité Security Hub. Les détails incluent le titre du contrôle, la description, la disponibilité de la région, les paramètres et d'autres informations.

```
aws securityhub get-security-control-definition \
    --security-control-id ACM.1
```

Sortie :

```
{
    "SecurityControlDefinition": {
        "SecurityControlId": "ACM.1",
        "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
        "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/ACM.1/
remediation",
        "SeverityRating": "MEDIUM",
        "CurrentRegionAvailability": "AVAILABLE",
        "ParameterDefinitions": {
            "daysToExpiration": {
                "Description": "Number of days within which the ACM certificate must
 be renewed",
                "ConfigurationOptions": {
                    "Integer": {
                        "DefaultValue": 30,
                        "Min": 14,
                        "Max": 365
                    }
                }
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Paramètres de contrôle personnalisés dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous GetSecurityControlDefinitionà la section Référence des AWS CLI commandes.

`invite-members`

L'exemple de code suivant montre comment utiliser`invite-members`.

AWS CLI

Pour envoyer des invitations aux comptes des membres

L'`invite-members`exemple suivant envoie des invitations aux comptes de membres spécifiés.

```
aws securityhub invite-members \
    --account-ids "123456789111" "123456789222"
```

Sortie :

```
{
    "UnprocessedAccounts": []
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous InviteMembersà la section Référence des AWS CLI commandes.

`list-automation-rules`

L'exemple de code suivant montre comment utiliser`list-automation-rules`.

AWS CLI

Pour consulter la liste des règles d'automatisation

L'`list-automation-rules`exemple suivant répertorie les règles d'automatisation d'un AWS compte. Seul le compte administrateur du Security Hub peut exécuter cette commande.

```
aws securityhub list-automation-rules \
    --max-results 3 \
    --next-token NULL
```

Sortie :

```
{
    "AutomationRulesMetadata": [
        {
```

```
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "Suppress informational findings",
            "Description": "Suppress GuardDuty findings with Informational
 severity",
            "IsTerminal": false,
            "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
            "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        },
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "sample rule",
            "Description": "A sample rule",
            "IsTerminal": false,
            "CreatedAt": "2023-07-15T23:37:20.223000+00:00",
            "UpdatedAt": "2023-07-15T23:37:20.223000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        },
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "sample rule",
            "Description": "A sample rule",
            "IsTerminal": false,
            "CreatedAt": "2023-07-15T23:45:25.126000+00:00",
            "UpdatedAt": "2023-07-15T23:45:25.126000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        }
    ]
}
```


Pour plus d'informations, consultez la section Affichage des règles d'automatisation dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListAutomationRulesà la section Référence des AWS CLI commandes.

`list-configuration-policies`

L'exemple de code suivant montre comment utiliser`list-configuration-policies`.

AWS CLI

Pour répertorier les résumés des politiques de configuration

L'`list-configuration-policies`exemple suivant présente un résumé des politiques de configuration de l'organisation.

```
aws securityhub list-configuration-policies \
    --max-items 3
```

Sortie :

```
{
    "ConfigurationPolicySummaries": [
        {
            "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Name": "SampleConfigurationPolicy1",
            "Description": "SampleDescription1",
            "UpdatedAt": "2023-09-26T21:08:36.214000+00:00",
            "ServiceEnabled": true
        },
        {
            "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Name": "SampleConfigurationPolicy2",
            "Description": "SampleDescription2"
            "UpdatedAt": "2023-11-28T19:26:25.207000+00:00",
            "ServiceEnabled": true
        },
        {
            "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Name": "SampleConfigurationPolicy3",
            "Description": "SampleDescription3",
            "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
```

```
            "ServiceEnabled": true
        }
}
```

Pour plus d'informations, consultez la section Visualisation des politiques de configuration de Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListConfigurationPoliciesà la section Référence des AWS CLI commandes.

`list-configuration-policy-associations`

L'exemple de code suivant montre comment utiliser`list-configuration-policy-associations`.

AWS CLI

Pour répertorier les associations de configuration

L'`list-configuration-policy-associations`exemple suivant répertorie un résumé des associations de configuration pour l'organisation. La réponse inclut des associations avec des politiques de configuration et des comportements autogérés.

```
aws securityhub list-configuration-policy-associations \
    --filters '{"AssociationType": "APPLIED"}' \
    --max-items 4
```

Sortie :

```
{
    "ConfigurationPolicyAssociationSummaries": [
        {
            "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TargetId": "r-1ab2",
            "TargetType": "ROOT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-11-28T19:26:49.417000+00:00",
            "AssociationStatus": "FAILED",
            "AssociationStatusMessage": "Policy association failed because 2
 organizational units or accounts under this root failed."
        },
```

```
        {
            "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "TargetId": "ou-1ab2-c3de4f5g",
            "TargetType": "ORGANIZATIONAL_UNIT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-09-26T21:14:05.283000+00:00",
            "AssociationStatus": "FAILED",
            "AssociationStatusMessage": "One or more children under this target
 failed association."
        },
        {
            "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "TargetId": "ou-6hi7-8j91kl2m",
            "TargetType": "ORGANIZATIONAL_UNIT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
            "AssociationStatus": "SUCCESS",
            "AssociationStatusMessage": "Association applied successfully on this
 target."
        },
        {
            "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
            "TargetId": "111122223333",
            "TargetType": "ACCOUNT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-11-28T22:01:26.409000+00:00",
            "AssociationStatus": "SUCCESS"
    }
}
```

Pour plus d'informations, consultez la section Affichage de l'état et des détails de la politique de configuration dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListConfigurationPolicyAssociationsà la section Référence des AWS CLI commandes.

`list-enabled-products-for-import`

L'exemple de code suivant montre comment utiliser`list-enabled-products-for-import`.

AWS CLI

Pour renvoyer la liste des intégrations de produits activées

L'`list-enabled-products-for-import`exemple suivant renvoie la liste des ARNS d'abonnement pour les intégrations de produits actuellement activées.

```
aws securityhub list-enabled-products-for-import
```

Sortie :

```
{
    "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-west-1:123456789012:product-subscription/aws/securityhub" ]
}
```

Pour plus d'informations, consultez la section Gestion des intégrations de produits dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListEnabledProductsForImportà la section Référence des AWS CLI commandes.

`list-finding-aggregators`

L'exemple de code suivant montre comment utiliser`list-finding-aggregators`.

AWS CLI

Pour répertorier les widgets disponibles

L'`list-finding-aggregators`exemple suivant renvoie l'ARN de la configuration d'agrégation de recherche.

```
aws securityhub list-finding-aggregators
```

Sortie :

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000"
}
```

Pour plus d'informations, consultez la section Affichage de la configuration actuelle de l'agrégation des résultats dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListFindingAggregatorsà la section Référence des AWS CLI commandes.

`list-invitations`

L'exemple de code suivant montre comment utiliser`list-invitations`.

AWS CLI

Pour afficher une liste d'invitations

L'`list-invitations`exemple suivant permet de récupérer la liste des invitations envoyées au compte demandeur.

```
aws securityhub list-invitations
```

Sortie :

```
{
    "Invitations": [
        {
            "AccountId": "123456789012",
            "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
            "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
            "MemberStatus": "ASSOCIATED"
        }
    ],
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListInvitationsà la section Référence des AWS CLI commandes.

`list-members`

L'exemple de code suivant montre comment utiliser`list-members`.

AWS CLI

Pour récupérer la liste des comptes des membres

L'`list-members`exemple suivant renvoie la liste des comptes membres pour le compte administrateur demandeur.

```
aws securityhub list-members
```

Sortie :

```
{
    "Members": [
        {
            "AccountId": "123456789111",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        },
        {
            "AccountId": "123456789222",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        }
    ],
}
```

Pour plus d'informations, consultez la section Gestion des comptes d'administrateur et de membre dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListMembersà la section Référence des AWS CLI commandes.

`list-organization-admin-accounts`

L'exemple de code suivant montre comment utiliser`list-organization-admin-accounts`.

AWS CLI

Pour répertorier les comptes d'administrateur Security Hub désignés

L'`list-organization-admin-accounts`exemple suivant répertorie les comptes d'administrateur du Security Hub d'une organisation.

```
aws securityhub list-organization-admin-accounts
```

Sortie :

```
{
    AdminAccounts": [
        { "AccountId": "777788889999" },
        { "Status": "ENABLED" }
    ]
}
```

Pour plus d'informations, consultez la section Désignation d'un compte administrateur Security Hub dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListOrganizationAdminAccountsà la section Référence des AWS CLI commandes.

`list-security-control-definitions`

L'exemple de code suivant montre comment utiliser`list-security-control-definitions`.

AWS CLI

Exemple 1 : pour répertorier tous les contrôles de sécurité disponibles

L'`list-security-control-definitions`exemple suivant répertorie les contrôles de sécurité disponibles dans toutes les normes du Security Hub. Cet exemple limite les résultats à trois contrôles.

```
aws securityhub list-security-control-definitions \
    --max-items 3
```

Sortie :

```
{
    "SecurityControlDefinitions": [
        {
            "SecurityControlId": "ACM.1",
```

```
            "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
            "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.1/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": [
                "Parameters"
            ]
        },
        {
            "SecurityControlId": "ACM.2",
            "Title": "RSA certificates managed by ACM should use a key length of at
 least 2,048 bits",
            "Description": "This control checks whether RSA certificates managed by
 AWS Certificate Manager use a key length of at least 2,048 bits. The control fails
 if the key length is smaller than 2,048 bits.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.2/remediation",
            "SeverityRating": "HIGH",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        },
        {
            "SecurityControlId": "APIGateway.1",
            "Title": "API Gateway REST and WebSocket API execution logging should be
 enabled",
            "Description": "This control checks whether all stages of an Amazon
 API Gateway REST or WebSocket API have logging enabled. The control fails if
 the 'loggingLevel' isn't 'ERROR' or 'INFO' for all stages of the API. Unless you
 provide custom parameter values to indicate that a specific log type should be
 enabled, Security Hub produces a passed finding if the logging level is either
 'ERROR' or 'INFO'.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
APIGateway.1/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": [
```

```
                "Parameters"
            ]
        }
    ],
    "NextToken": "U2FsdGVkX1/UprCPzxVbkDeHikDXbDxfgJZ1w2RG1XWsFPTMTIQPVE0m/
FduIGxS7ObRtAbaUt/8/RCQcg2PU0YXI20hH/GrhoOTgv+TSm0qvQVFhkJepWmqh
+NYawjocVBeos6xzn/8qnbF9IuwGg=="
}
```

Pour plus d'informations, consultez la section Affichage des détails d'une norme dans le Guide de l'utilisateur du AWS Security Hub.

Exemple 2 : Pour répertorier les contrôles de sécurité disponibles pour une norme spécifique

L'`list-security-control-definitions`exemple suivant répertorie les contrôles de sécurité disponibles pour le CIS AWS Foundations Benchmark v1.4.0. Cet exemple limite les résultats à trois contrôles.

```
aws securityhub list-security-control-definitions \
    --standards-arn "arn:aws:securityhub:us-east-1::standards/cis-aws-foundations-
benchmark/v/1.4.0" \
    --max-items 3
```

Sortie :

```
{
    "SecurityControlDefinitions": [
        {
            "SecurityControlId": "CloudTrail.1",
            "Title": "CloudTrail should be enabled and configured with at least one
 multi-Region trail that includes read and write management events",
            "Description": "This AWS control checks that there is at least one
 multi-region AWS CloudTrail trail includes read and write management events.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.1/remediation",
            "SeverityRating": "HIGH",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        },
        {
            "SecurityControlId": "CloudTrail.2",
            "Title": "CloudTrail should have encryption at-rest enabled",
```

```
            "Description": "This AWS control checks whether AWS CloudTrail is
 configured to use the server side encryption (SSE) AWS Key Management Service (AWS
 KMS) customer master key (CMK) encryption. The check will pass if the KmsKeyId is
 defined.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.2/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        },
        {
            "SecurityControlId": "CloudTrail.4",
            "Title": "CloudTrail log file validation should be enabled",
            "Description": "This AWS control checks whether CloudTrail log file
 validation is enabled.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.4/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}
```

Pour plus d'informations, consultez la section Affichage des détails d'une norme dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListSecurityControlDefinitionsà la section Référence des AWS CLI commandes.

`list-standards-control-associations`

L'exemple de code suivant montre comment utiliser`list-standards-control-associations`.

AWS CLI

Pour obtenir le statut d'activation d'un contrôle dans chaque norme activée

L'`list-standards-control-associations`exemple suivant répertorie le statut d'activation de CloudTrail .1 dans chaque norme activée.

```
aws securityhub list-standards-control-associations \
```

```
    --security-control-id CloudTrail.1
```

Sortie :

```
{
    "StandardsControlAssociationSummaries": [
        {
            "StandardsArn": "arn:aws:securityhub:us-east-2::standards/nist-800-53/
v/5.0.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "NIST.800-53.r5 AC-2(4)",
                "NIST.800-53.r5 AC-4(26)",
                "NIST.800-53.r5 AC-6(9)",
                "NIST.800-53.r5 AU-10",
                "NIST.800-53.r5 AU-12",
                "NIST.800-53.r5 AU-2",
                "NIST.800-53.r5 AU-3",
                "NIST.800-53.r5 AU-6(3)",
                "NIST.800-53.r5 AU-6(4)",
                "NIST.800-53.r5 AU-14(1)",
                "NIST.800-53.r5 CA-7",
                "NIST.800-53.r5 SC-7(9)",
                "NIST.800-53.r5 SI-3(8)",
                "NIST.800-53.r5 SI-4(20)",
                "NIST.800-53.r5 SI-7(8)",
                "NIST.800-53.r5 SA-8(22)"
            ],
            "UpdatedAt": "2023-05-15T17:52:21.304000+00:00",
            "StandardsControlTitle": "CloudTrail should be enabled and configured
 with at least one multi-Region trail that includes read and write management
 events",
            "StandardsControlDescription": "This AWS control checks that there is
 at least one multi-region AWS CloudTrail trail includes read and write management
 events."
        },
        {
            "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
            "SecurityControlId": "CloudTrail.1",
```

```
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "CIS AWS Foundations 2.1"
            ],
            "UpdatedAt": "2020-02-10T21:22:53.998000+00:00",
            "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
            "StandardsControlDescription": "AWS CloudTrail is a web service that
 records AWS API calls for your account and delivers log files to you. The recorded
 information includes the identity of the API caller, the time of the API call,
 the source IP address of the API caller, the request parameters, and the response
 elements returned by the AWS service."
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-east-2::standards/aws-
foundational-security-best-practices/v/1.0.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "DISABLED",
            "RelatedRequirements": [],
            "UpdatedAt": "2023-05-15T19:31:52.671000+00:00",
            "UpdatedReason": "Alternative compensating controls are in place",
            "StandardsControlTitle": "CloudTrail should be enabled and configured
 with at least one multi-Region trail that includes read and write management
 events",
            "StandardsControlDescription": "This AWS control checks that there is
 at least one multi-region AWS CloudTrail trail includes read and write management
 events."
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-east-2::standards/cis-aws-
foundations-benchmark/v/1.4.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "CIS AWS Foundations Benchmark v1.4.0/3.1"
            ],
            "UpdatedAt": "2022-11-10T15:40:36.021000+00:00",
            "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
```

```
            "StandardsControlDescription": "AWS CloudTrail is a web service that
 records AWS API calls for your account and delivers log files to you. The recorded
 information includes the identity of the API caller, the time of the API call,
 the source IP address of the API caller, the request parameters, and the response
 elements returned by the AWS service. CloudTrail provides a history of AWS API
 calls for an account, including API calls made via the Management Console, SDKs,
 command line tools, and higher-level AWS services (such as CloudFormation)."
        }
    ]
}
```

Pour plus d'informations, consultez la section Activation et désactivation des contrôles dans des normes spécifiques dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous ListStandardsControlAssociationsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour récupérer les balises attribuées à une ressource

L'`list-tags-for-resource`exemple suivant renvoie les balises attribuées à la ressource hub spécifiée.

```
aws securityhub list-tags-for-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Sortie :

```
{
    "Tags": {
        "Department" : "Operations",
        "Area" : "USMidwest"
    }
}
```

Pour plus d'informations, consultez AWS: : SecurityHub : :Hub dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`start-configuration-policy-association`

L'exemple de code suivant montre comment utiliser`start-configuration-policy-association`.

AWS CLI

Exemple 1 : pour associer une politique de configuration

L'`start-configuration-policy-association`exemple suivant associe la politique de configuration spécifiée à l'unité organisationnelle spécifiée. Une configuration peut être associée à un compte cible, à une unité organisationnelle ou à la racine.

```
aws securityhub start-configuration-policy-association \
    --configuration-policy-identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

Sortie :

```
{
    "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "TargetId": "ou-6hi7-8j91kl2m",
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
    "AssociationStatus": "PENDING"
}
```

Pour plus d'informations, consultez la section Création et association de politiques de configuration de Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

Exemple 2 : pour associer une configuration autogérée

L'`start-configuration-policy-association`exemple suivant associe une configuration autogérée au compte spécifié.

```
aws securityhub start-configuration-policy-association \
```

```
    --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
    --target '{"OrganizationalUnitId": "123456789012"}'
```

Sortie :

```
{
    "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
    "TargetId": "123456789012",
    "TargetType": "ACCOUNT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
    "AssociationStatus": "PENDING"
}
```

Pour plus d'informations, consultez la section Création et association de politiques de configuration de Security Hub dans le Guide de l'utilisateur de AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous StartConfigurationPolicyAssociationà la section Référence des AWS CLI commandes.

`start-configuration-policy-disassociation`

L'exemple de code suivant montre comment utiliser`start-configuration-policy-disassociation`.

AWS CLI

Exemple 1 : pour dissocier une politique de configuration

L'`start-configuration-policy-disassociation`exemple suivant dissocie une politique de configuration de l'unité organisationnelle spécifiée. Une configuration peut être dissociée d'un compte cible, d'une unité organisationnelle ou de la racine.

```
aws securityhub start-configuration-policy-disassociation \
    --configuration-policy-identifier "arn:aws:securityhub:eu-
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Dissocier une configuration des comptes et OUs dans le Guide de l'utilisateur du AWS Security Hub.

Exemple 2 : pour dissocier une configuration autogérée

L'`start-configuration-policy-disassociation`exemple suivant dissocie une configuration autogérée du compte spécifié.

```
aws securityhub start-configuration-policy-disassociation \
    --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
    --target '{"AccountId": "123456789012"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Dissocier une configuration des comptes et OUs dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous StartConfigurationPolicyDisassociationà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour attribuer un tag à une ressource

L'`tag-resource`exemple suivant attribue des valeurs aux balises Department et Area à la ressource hub spécifiée.

```
aws securityhub tag-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \
    --tags '{"Department":"Operations", "Area":"USMidwest"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez AWS: : SecurityHub : :Hub dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer une valeur de balise d'une ressource

L'`untag-resource`exemple suivant supprime la balise Department de la ressource hub spécifiée.

```
aws securityhub untag-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \
    --tag-keys "Department"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez AWS: : SecurityHub : :Hub dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-action-target`

L'exemple de code suivant montre comment utiliser`update-action-target`.

AWS CLI

Pour mettre à jour une action personnalisée

L'`update-action-target`exemple suivant met à jour le nom de l'action personnalisée identifiée par l'ARN spécifié.

```
aws securityhub update-action-target \
    --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation" \
    --name "Send to remediation"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Création d'une action personnalisée et association de celle-ci à une règle d' CloudWatch événements dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateActionTargetà la section Référence des AWS CLI commandes.

`update-configuration-policy`

L'exemple de code suivant montre comment utiliser`update-configuration-policy`.

AWS CLI

Pour mettre à jour une politique de configuration

L'`update-configuration-policy`exemple suivant met à jour une politique de configuration existante afin d'utiliser les paramètres spécifiés.

```
aws securityhub update-configuration-policy \
    --identifier "arn:aws:securityhub:eu-central-1:508236694226:configuration-policy/09f37766-57d8-4ede-9d33-5d8b0fecf70e" \
    --name "SampleConfigurationPolicyUpdated" \
    --description "SampleDescriptionUpdated" \
    --configuration-policy '{"SecurityHub": {"ServiceEnabled": true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-central-1::standards/aws-foundational-security-best-practices/v/1.0.0","arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0"],"SecurityControlsConfiguration":{"DisabledSecurityControlIdentifiers": ["CloudWatch.1"], "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer": 21}}}}]}}}' \
    --updated-reason "Disabling CloudWatch.1 and changing parameter value"
```

Sortie :

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "SampleConfigurationPolicyUpdated",
    "Description": "SampleDescriptionUpdated",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
```

```
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudWatch.1"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 21
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Mise à jour des politiques de configuration de AWS Security Hub dans le Guide de l'utilisateur du Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateConfigurationPolicyà la section Référence des AWS CLI commandes.

`update-finding-aggregator`

L'exemple de code suivant montre comment utiliser`update-finding-aggregator`.

AWS CLI

Pour mettre à jour la configuration actuelle de l'agrégation des résultats

L'`update-finding-aggregator`exemple suivant modifie la configuration d'agrégation des recherches pour établir un lien à partir des régions sélectionnées. Il est géré depuis l'est des États-Unis (Virginie), qui est la région d'agrégation. Elle sélectionne l'ouest des États-Unis (Californie du Nord) et l'ouest des États-Unis (Oregon) comme régions liées.

```
aws securityhub update-finding-aggregator \
    --region us-east-1 \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000 \
    --region-linking-mode SPECIFIED_REGIONS \
    --regions us-west-1,us-west-2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Mise à jour de la configuration d'agrégation des résultats dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateFindingAggregatorà la section Référence des AWS CLI commandes.

`update-insight`

L'exemple de code suivant montre comment utiliser`update-insight`.

AWS CLI

Exemple 1 : pour modifier le filtre d'un aperçu personnalisé

L'`update-insight`exemple suivant modifie les filtres pour obtenir un aperçu personnalisé. Les informations mises à jour recherchent les résultats très graves liés aux AWS rôles.

```
aws securityhub update-insight \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
    --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' \
    --name "High severity role findings"
```

Exemple 2 : pour modifier l'attribut de regroupement pour un aperçu personnalisé

L'`update-insight`exemple suivant modifie l'attribut de regroupement pour l'aperçu personnalisé avec l'ARN spécifié. Le nouvel attribut de regroupement est l'ID de ressource.

```
aws securityhub update-insight \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
    --group-by-attribute "ResourceId" \
    --name "Critical role findings"
```

Sortie :

```
{
    "Insights": [
        {
            "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
            "Name": "Critical role findings",
            "Filters": {
                "SeverityLabel": [
                    {
                        "Value": "CRITICAL",
                        "Comparison": "EQUALS"
                    }
                ],
                "ResourceType": [
                    {
                        "Value": "AwsIamRole",
                        "Comparison": "EQUALS"
                    }
                ]
            },
            "GroupByAttribute": "ResourceId"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des informations personnalisées dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateInsightà la section Référence des AWS CLI commandes.

`update-organization-configuration`

L'exemple de code suivant montre comment utiliser`update-organization-configuration`.

AWS CLI

Pour mettre à jour la configuration de Security Hub pour une organisation

L'`update-organization-configuration`exemple suivant indique que Security Hub doit utiliser la configuration centrale pour configurer une organisation. Après avoir exécuté cette commande, l'administrateur délégué du Security Hub peut créer et gérer des politiques de configuration pour configurer l'organisation. L'administrateur délégué peut également utiliser cette commande pour passer de la configuration centrale à la configuration locale. Si la configuration locale est le type de configuration, l'administrateur délégué peut choisir d'activer automatiquement Security Hub et les normes de sécurité par défaut dans les nouveaux comptes de l'organisation.

```
aws securityhub update-organization-configuration \
    --no-auto-enable \
    --organization-configuration '{"ConfigurationType": "CENTRAL"}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes auprès d' AWS Organizations dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateOrganizationConfigurationà la section Référence des AWS CLI commandes.

`update-security-control`

L'exemple de code suivant montre comment utiliser`update-security-control`.

AWS CLI

Pour mettre à jour les propriétés du contrôle de sécurité

L'`update-security-control`exemple suivant indique des valeurs personnalisées pour un paramètre de contrôle de sécurité du Security Hub.

```
aws securityhub update-security-control \
    --security-control-id ACM.1 \
    --parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":
 15}}}' \
    --last-update-reason "Internal compliance requirement"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Paramètres de contrôle personnalisés dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateSecurityControlà la section Référence des AWS CLI commandes.

`update-security-hub-configuration`

L'exemple de code suivant montre comment utiliser`update-security-hub-configuration`.

AWS CLI

Pour mettre à jour la configuration de Security Hub

L'`update-security-hub-configuration`exemple suivant configure Security Hub pour activer automatiquement de nouvelles commandes pour les normes activées.

```
aws securityhub update-security-hub-configuration \
    --auto-enable-controls
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Activation automatique des nouvelles commandes dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateSecurityHubConfigurationà la section Référence des AWS CLI commandes.

`update-standards-control`

L'exemple de code suivant montre comment utiliser`update-standards-control`.

AWS CLI

Exemple 1 : pour désactiver un contrôle

L'`update-standards-control`exemple suivant désactive le PCI. AutoScaling1. Contrôle.

```
aws securityhub update-standards-control \
    --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1" \
    --control-status "DISABLED" \
    --disabled-reason "Not applicable for my service"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour activer un contrôle

L'`update-standards-control`exemple suivant active le PCI. AutoScaling1. Contrôle.

```
aws securityhub update-standards-control \
    --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1" \
    --control-status "ENABLED"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation et activation des contrôles individuels dans le Guide de l'utilisateur du AWS Security Hub.

- Pour plus de détails sur l'API, reportez-vous UpdateStandardsControlà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Security Lake AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Security Lake.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-aws-log-source`

L'exemple de code suivant montre comment utiliser`create-aws-log-source`.

AWS CLI

Pour ajouter un Amazon Web Service pris en charge de manière native en tant que source Amazon Security Lake

L'`create-aws-logsource`exemple suivant ajoute les journaux de flux VPC en tant que source Security Lake dans les comptes et régions désignés.

```
aws securitylake create-aws-log-source \
    --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
 "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

Sortie :

```
{
    "failed": [
        "123456789012"
    ]
}
```

Pour plus d'informations, consultez la section Ajout d'un AWS service en tant que source dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateAwsLogSourceà la section Référence des AWS CLI commandes.

`create-custom-log-source`

L'exemple de code suivant montre comment utiliser`create-custom-log-source`.

AWS CLI

Pour ajouter une source personnalisée en tant que source Amazon Security Lake

L'`create-custom-logsource`exemple suivant ajoute une source personnalisée en tant que source Security Lake dans le compte du fournisseur de journaux désigné et dans la région désignée.

```
aws securitylake create-custom-log-source \
    --source-name "VPC_FLOW" \
    --event-classes '["DNS_ACTIVITY", "NETWORK_ACTIVITY"]' \
    --configuration '{"crawlerConfiguration": {"roleArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4"},"providerIdentity": {"principal":
 "029189416600","externalId": "123456789012"}}' --region "us-east-1"
```

Sortie :

```
{
    "customLogSource": {
        "attributes": {
            "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRXT0D4",
            "databaseArn": "arn:aws:glue:eu-west-2:123456789012:database/
E1WG1ZNPRXT0D4",
            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/E1WG1ZNPRXT0D4"
        },
        "provider": {
            "location": "amzn-s3-demo-bucket--usw2-az1--x-s3",
            "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-Provider-
testCustom2-eu-west-2"
        },
        "sourceName": "testCustom2"
        "sourceVersion": "2.0"
    }
}
```

Pour plus d'informations, consultez la section Ajout d'une source personnalisée dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateCustomLogSourceà la section Référence des AWS CLI commandes.

`create-data-lake-exception-subscription`

L'exemple de code suivant montre comment utiliser`create-data-lake-exception-subscription`.

AWS CLI

Pour envoyer des notifications concernant les exceptions de Security Lake

L'`create-data-lake-exception-subscription`exemple suivant envoie des notifications relatives aux exceptions de Security Lake au compte spécifié par SMS. Le message d'exception est conservé pendant la période spécifiée.

```
aws securitylake create-data-lake-exception-subscription \
    --notification-endpoint "123456789012" \
    --exception-time-to-live 30 \
    --subscription-protocol "sms"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Résolution des problèmes liés à Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateDataLakeExceptionSubscriptionà la section Référence des AWS CLI commandes.

`create-data-lake-organization-configuration`

L'exemple de code suivant montre comment utiliser`create-data-lake-organization-configuration`.

AWS CLI

Pour configurer Security Lake dans les nouveaux comptes d'organisation

L'`create-data-lake-organization-configuration`exemple suivant active Security Lake et la collecte des événements sources spécifiés et enregistre les nouveaux comptes d'organisation.

```
aws securitylake create-data-lake-organization-configuration \
    --auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS","sourceVersion": "1.0"}]}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de plusieurs comptes auprès d' AWS Organizations dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateDataLakeOrganizationConfigurationà la section Référence des AWS CLI commandes.

`create-data-lake`

L'exemple de code suivant montre comment utiliser`create-data-lake`.

AWS CLI

Exemple 1 : pour configurer votre lac de données dans plusieurs régions

L'`create-data-lake`exemple suivant active Amazon Security Lake dans plusieurs AWS régions et configure votre lac de données.

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

Sortie :

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
```

```
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        },
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-2",
```

```
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

Exemple 2 : pour configurer votre lac de données dans une seule région

L'`create-data-lake`exemple suivant active Amazon Security Lake dans une seule AWS région et configure votre lac de données.

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

Sortie :

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
```

```
            "encryptionConfiguration": {
                "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 500
                },
                "transitions": [
                    {
                        "days": 30,
                        "storageClass": "GLACIER"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateDataLakeà la section Référence des AWS CLI commandes.

`create-subscriber-notification`

L'exemple de code suivant montre comment utiliser`create-subscriber-notification`.

AWS CLI

Pour créer une notification d'abonné

L'`create-subscriber-notification`exemple suivant montre comment spécifier une notification aux abonnés pour créer une notification lorsque de nouvelles données sont écrites dans le lac de données.

```
aws securitylake create-subscriber-notification \
    --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
    --configuration '{"httpsNotificationConfiguration":
 {"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",
 "endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

Sortie :

```
{
    "subscriberEndpoint": [
        "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
    ]
}
```

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateSubscriberNotificationà la section Référence des AWS CLI commandes.

`create-subscriber`

L'exemple de code suivant montre comment utiliser`create-subscriber`.

AWS CLI

Exemple 1 : pour créer un abonné avec accès aux données

L'`create-subscriber`exemple suivant crée un abonné dans Security Lake avec accès aux données de la AWS région actuelle pour l'identité d'abonné spécifiée pour une AWS source.

```
aws securitylake create-subscriber \
    --access-types "S3" \
```

```
    --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW","sourceVersion":
 "2.0"}}]' \
    --subscriber-name 'opensearch-s3' \
    --subscriber-identity '{"principal": "029189416600","externalId":
 "123456789012"}'
```

Sortie :

```
{
    "subscriber": {
        "accessTypes": [
            "S3"
        ],
        "createdAt": "2024-07-17T19:08:26.787000+00:00",
        "roleArn": "arn:aws:iam::773172568199:role/AmazonSecurityLake-896f218b-
cfba-40be-a255-8b49a65d0407",
        "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
um632ufwpvxkyz0bc5hkb64atycnf3",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "2.0"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:us-
east-1:773172568199:subscriber/896f218b-cfba-40be-a255-8b49a65d0407",
        "subscriberId": "896f218b-cfba-40be-a255-8b49a65d0407",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "029189416600"
        },
        "subscriberName": "opensearch-s3",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-17T19:08:27.133000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Création d'un abonné avec accès aux données dans le guide de l'utilisateur d'Amazon Security Lake.

Exemple 2 : pour créer un abonné avec accès aux requêtes

L'`create-subscriber`exemple suivant crée un abonné dans Security Lake avec un accès aux requêtes dans la AWS région actuelle pour l'identité d'abonné spécifiée.

```
aws securitylake create-subscriber \
    --access-types "LAKEFORMATION" \
    --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW","sourceVersion":
 "2.0"}}]' \
    --subscriber-name 'opensearch-s3' \
    --subscriber-identity '{"principal": "029189416600","externalId":
 "123456789012"}'
```

Sortie :

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-07-18T01:05:55.853000+00:00",
        "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/8c31da49-c224-4f1e-bb12-37ab756d6d8a",
        "resourceShareName": "LakeFormation-V2-NAMENAMENA-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "2.0"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:us-east-1:123456789012:subscriber/
e762aabb-ce3d-4585-beab-63474597845d",
        "subscriberId": "e762aabb-ce3d-4585-beab-63474597845d",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "029189416600"
        },
        "subscriberName": "opensearch-s3",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-18T01:05:58.393000+00:00"
    }
}
```

Pour plus d'informations, consultez la section Création d'un abonné avec accès aux requêtes dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous CreateSubscriberà la section Référence des AWS CLI commandes.

`delete-aws-log-source`

L'exemple de code suivant montre comment utiliser`delete-aws-log-source`.

AWS CLI

Pour supprimer un service pris en charge de manière native. AWS

L'`delete-aws-logsource`exemple suivant supprime les journaux de flux VPC en tant que source Security Lake dans les comptes et régions désignés.

```
aws securitylake delete-aws-log-source \
    --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
 "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

Sortie :

```
{
    "failed": [
        "123456789012"
    ]
}
```

Pour plus d'informations, consultez la section Suppression d'un AWS service en tant que source dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous DeleteAwsLogSourceà la section Référence des AWS CLI commandes.

`delete-custom-log-source`

L'exemple de code suivant montre comment utiliser`delete-custom-log-source`.

AWS CLI

Pour supprimer une source personnalisée.

L'`delete-custom-logsource`exemple suivant supprime une source personnalisée dans le compte du fournisseur de journaux désigné dans la région désignée.

```
aws securitylake delete-custom-log-source \
    --source-name "CustomSourceName"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Suppression d'une source personnalisée dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous DeleteCustomLogSourceà la section Référence des AWS CLI commandes.

`delete-data-lake-organization-configuration`

L'exemple de code suivant montre comment utiliser`delete-data-lake-organization-configuration`.

AWS CLI

Pour arrêter la collecte automatique des sources dans les comptes des membres

L'`delete-data-lake-organization-configuration`exemple suivant arrête la collecte automatique des résultats du AWS Security Hub à partir des nouveaux comptes membres qui rejoignent l'organisation. Seul l'administrateur délégué de Security Lake peut exécuter cette commande. Cela empêche les nouveaux comptes membres de fournir automatiquement des données au lac de données.

```
aws securitylake delete-data-lake-organization-configuration \
    --auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS"}]}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de plusieurs comptes auprès d' AWS Organizations dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous DeleteDataLakeOrganizationConfigurationà la section Référence des AWS CLI commandes.

`delete-data-lake`

L'exemple de code suivant montre comment utiliser`delete-data-lake`.

AWS CLI

Pour désactiver votre lac de données

L'`delete-data-lake`exemple suivant désactive votre lac de données dans les AWS régions spécifiées. Dans les régions spécifiées, les sources ne fournissent plus de données au lac de données. Pour un déploiement de Security Lake utilisant AWS Organizations, seul l'administrateur délégué de Security Lake à l'organisation peut désactiver Security Lake pour les comptes de l'organisation.

```
aws securitylake delete-data-lake \
    --regions "ap-northeast-1" "eu-central-1"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désactivation d'Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous DeleteDataLakeà la section Référence des AWS CLI commandes.

`delete-subscriber-notification`

L'exemple de code suivant montre comment utiliser`delete-subscriber-notification`.

AWS CLI

Pour supprimer une notification d'abonné

L'`delete-subscriber-notification`exemple suivant montre comment supprimer la notification d'abonné pour un abonné Security Lake spécifique.

```
aws securitylake delete-subscriber-notification \
    --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriberNotificationà la section Référence des AWS CLI commandes.

`delete-subscriber`

L'exemple de code suivant montre comment utiliser`delete-subscriber`.

AWS CLI

Pour supprimer un abonné

L'`delete-subscriber`exemple suivant montre comment supprimer un abonné si vous ne souhaitez plus qu'il consomme les données de Security Lake.

```
aws securitylake delete-subscriber \
    --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriberà la section Référence des AWS CLI commandes.

`get-data-lake-exception-subscription`

L'exemple de code suivant montre comment utiliser`get-data-lake-exception-subscription`.

AWS CLI

Pour obtenir des informations sur un abonnement exceptionnel

L'`get-data-lake-exception-subscription`exemple suivant fournit des détails sur un abonnement d'exception à Security Lake. Dans cet exemple, l'utilisateur du AWS compte spécifié est informé des erreurs par l'envoi de SMS. Le message d'exception reste dans le compte pendant la période spécifiée. Un abonnement d'exception informe un utilisateur de Security Lake d'une erreur via le protocole préféré du demandeur.

```
aws securitylake get-data-lake-exception-subscription
```

Sortie :

```
{
    "exceptionTimeToLive": 30,
    "notificationEndpoint": "123456789012",
    "subscriptionProtocol": "sms"
}
```

Pour plus d'informations, consultez la section Résolution des problèmes liés à l'état du lac de données dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous GetDataLakeExceptionSubscriptionà la section Référence des AWS CLI commandes.

`get-data-lake-organization-configuration`

L'exemple de code suivant montre comment utiliser`get-data-lake-organization-configuration`.

AWS CLI

Pour obtenir des informations détaillées sur la configuration des nouveaux comptes d'organisation

L'`get-data-lake-organization-configuration`exemple suivant permet de récupérer des informations sur les journaux sources que les nouveaux comptes d'entreprise enverront après leur intégration à Amazon Security Lake.

```
aws securitylake get-data-lake-organization-configuration
```

Sortie :

```
{
    "autoEnableNewAccount": [
        {
            "region": "us-east-1",
            "sources": [
                {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
```

```
                },
                {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                },
                {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion de plusieurs comptes auprès d' AWS Organizations dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous GetDataLakeOrganizationConfigurationà la section Référence des AWS CLI commandes.

`get-data-lake-sources`

L'exemple de code suivant montre comment utiliser`get-data-lake-sources`.

AWS CLI

Pour connaître l'état de la collecte des journaux

L'`get-data-lake-sources`exemple suivant permet d'obtenir un instantané de la collecte de journaux pour le compte spécifié dans la AWS région actuelle. Amazon Security Lake est activé sur le compte.

```
aws securitylake get-data-lake-sources \
    --accounts "123456789012"
```

Sortie :

```
{
    "dataLakeSources": [
        {
            "account": "123456789012",
            "sourceName": "SH_FINDINGS",
```

```
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "VPC_FLOW",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "NOT_COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "LAMBDA_EXECUTION",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "ROUTE53",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "CLOUD_TRAIL_MGMT",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
```

```
                }
            ]
        }
    ],
    "dataLakeArn": null
}
```

Pour plus d'informations, consultez la section Collecte de données à partir AWS des services dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous GetDataLakeSourcesà la section Référence des AWS CLI commandes.

`get-subscriber`

L'exemple de code suivant montre comment utiliser`get-subscriber`.

AWS CLI

Pour récupérer les informations d'abonnement

L'`get-subscriber`exemple suivant récupère les informations d'abonnement pour l'abonné Securiy Lake spécifié.

```
aws securitylake get-subscriber \
    --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-04-19T15:19:44.421803+00:00",
        "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "LAMBDA_EXECUTION",
                    "sourceVersion": "1.0"
```

```
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "EKS_AUDIT",
                    "sourceVersion": "2.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
testCustom2",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom2"
                    },
                    "provider": {
                        "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjbldwbgf4vge98av9/ext/testCustom2/",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-testCustom2-eu-west-2"
                    },
                    "sourceName": "testCustom2"
                }
```

```
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
TestCustom",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom"
                    },
                    "provider": {
                        "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjbldwbgf4vge98av9/ext/TestCustom/",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-TestCustom-eu-west-2"
                    },
                    "sourceName": "TestCustom"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "123456789012"
        },
        "subscriberName": "test",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-04-19T15:19:55.230588+00:00"
    }
}
```

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous GetSubscriberà la section Référence des AWS CLI commandes.

`list-data-lake-exceptions`

L'exemple de code suivant montre comment utiliser`list-data-lake-exceptions`.

AWS CLI

Pour répertorier les problèmes affectant votre lac de données

L'`list-data-lake-exceptions`exemple suivant répertorie les problèmes qui ont affecté votre lac de données au cours des 14 derniers jours dans les AWS régions spécifiées.

```
aws securitylake list-data-lake-exceptions \
    --regions "us-east-1" "eu-west-3"
```

Sortie :

```
{
    "exceptions": [
        {
            "exception": "The account does not have the required role permissions.
 Update your role permissions to use the new data source version.",
            "region": "us-east-1",
            "timestamp": "2024-02-29T12:24:15.641725+00:00"
        },
        {
            "exception": "The account does not have the required role permissions.
 Update your role permissions to use the new data source version.",
            "region": "eu-west-3",
            "timestamp": "2024-02-29T12:24:15.641725+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Résolution des problèmes liés à Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous ListDataLakeExceptionsà la section Référence des AWS CLI commandes.

`list-data-lakes`

L'exemple de code suivant montre comment utiliser`list-data-lakes`.

AWS CLI

Pour répertorier l'objet de configuration Security Lake

L'`list-data-lakes`exemple suivant répertorie l'objet de configuration Amazon Security Lake pour la AWS région spécifiée. Vous pouvez utiliser cette commande pour déterminer si Security Lake est activé dans une ou plusieurs régions spécifiques.

```
aws securitylake list-data-lakes \
    --regions "us-east-1"
```

Sortie :

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:123456789012:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:123456789012:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-
east-1-1234567890abcdef0",
            "updateStatus": {
                "exception": {
                    "code": "software.amazon.awssdk.services.s3.model.S3Exception",
                    "reason": ""
```

```
                },
                "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "status": "FAILED"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Vérification du statut des régions dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous ListDataLakesà la section Référence des AWS CLI commandes.

`list-log-sources`

L'exemple de code suivant montre comment utiliser`list-log-sources`.

AWS CLI

Pour récupérer les sources du journal Amazon Security Lake

L'`list-log-sources`exemple suivant répertorie les sources de log Amazon Security Lake d'un compte spécifié.

```
aws securitylake list-log-sources \
    --accounts "123456789012"
```

Sortie :

```
{
    "account": "123456789012",
    "region": "xy-region-1",
    "sources": [
        {
               "awsLogSource": {
                "sourceName": "VPC_FLOW",
                "sourceVersion": "2.0"
            }
        },
        {
            "awsLogSource": {
                "sourceName": "SH_FINDINGS",
```

```
                "sourceVersion": "2.0"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des sources dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous ListLogSourcesà la section Référence des AWS CLI commandes.

`list-subscribers`

L'exemple de code suivant montre comment utiliser`list-subscribers`.

AWS CLI

Pour récupérer les abonnés Amazon Security Lake

L'`list-subscribers`exemple suivant répertorie tous les abonnés Amazon Security Lake d'un compte spécifique.

```
aws securitylake list-subscribers
```

Sortie :

```
{
    "subscribers": [
        {
            "accessTypes": [
                "S3"
            ],
            "createdAt": "2024-06-04T15:02:28.921000+00:00",
            "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-
E1WG1ZNPRXT0D4",
            "s3BucketArn": "amzn-s3-demo-bucket--usw2-az1--x-s3",
            "sources": [
                {
                    "awsLogSource": {
                        "sourceName": "CLOUD_TRAIL_MGMT",
                        "sourceVersion": "2.0"
                    }
```

```
                },
                {
                    "awsLogSource": {
                        "sourceName": "LAMBDA_EXECUTION",
                        "sourceVersion": "1.0"
                    }
                },
                {
                    "customLogSource": {
                        "attributes": {
                            "crawlerArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4",
                            "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
                        },
                        "provider": {
                            "location": "amzn-s3-demo-bucket--usw2-az1--x-s3",
                            "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
                        },
                        "sourceName": "testCustom2"
                    }
                }
            ],
            "subscriberArn": "arn:aws:securitylake:eu-
west-2:123456789012:subscriber/E1WG1ZNPRXT0D4",
            "subscriberEndpoint": "arn:aws:sqs:eu-
west-2:123456789012:AmazonSecurityLake-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-Main-
Queue",
            "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "subscriberIdentity": {
                "externalId": "ext123456789012",
                "principal": "123456789012"
            },
            "subscriberName": "Test",
            "subscriberStatus": "ACTIVE",
            "updatedAt": "2024-06-04T15:02:35.617000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous ListSubscribersà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource existante

L'`list-tags-for-resource`exemple suivant répertorie les balises pour l'abonné Amazon Security Lake spécifié. Dans cet exemple, aucune valeur de balise n'est associée à la clé de balise Owner. Vous pouvez également utiliser cette opération pour répertorier les balises d'autres ressources Security Lake existantes.

```
aws securitylake list-tags-for-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab"
```

Sortie :

```
{
    "tags": [
        {
            "key": "Environment",
            "value": "Cloud"
        },
        {
            "key": "CostCenter",
            "value": "12345"
        },
        {
            "key": "Owner",
            "value": ""
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage des ressources Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`register-data-lake-delegated-administrator`

L'exemple de code suivant montre comment utiliser`register-data-lake-delegated-administrator`.

AWS CLI

Pour désigner l'administrateur délégué

L'`register-data-lake-delegated-administrator`exemple suivant désigne le AWS compte spécifié en tant qu'administrateur délégué d'Amazon Security Lake.

```
aws securitylake register-data-lake-delegated-administrator \
    --account-id 123456789012
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion de plusieurs comptes auprès d' AWS Organizations dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous RegisterDataLakeDelegatedAdministratorà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une ressource existante

L'`tag-resource`exemple suivant ajoute des balises à une ressource d'abonné existante. Pour créer une nouvelle ressource et y ajouter une ou plusieurs balises, n'utilisez pas cette opération. Utilisez plutôt l'opération de création appropriée pour le type de ressource que vous souhaitez créer.

```
aws securitylake tag-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --tags key=Environment,value=Cloud
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource existante

L'`untag-resource`exemple suivant supprime les balises spécifiées d'une ressource d'abonné existante.

```
aws securitylake untag-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --tags Environment Owner
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage des ressources Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-data-lake-exception-subscription`

L'exemple de code suivant montre comment utiliser`update-data-lake-exception-subscription`.

AWS CLI

Pour mettre à jour l'abonnement aux notifications pour les exceptions de Security Lake

L'`update-data-lake-exception-subscription`exemple suivant met à jour l'abonnement aux notifications qui informe les utilisateurs des exceptions de Security Lake.

```
aws securitylake update-data-lake-exception-subscription \
    --notification-endpoint "123456789012" \
    --exception-time-to-live 30 \
    --subscription-protocol "email"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Résolution des problèmes liés à Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous UpdateDataLakeExceptionSubscriptionà la section Référence des AWS CLI commandes.

`update-data-lake`

L'exemple de code suivant montre comment utiliser`update-data-lake`.

AWS CLI

Exemple 1 : pour mettre à jour les paramètres de votre lac de données

L'`update-data-lake`exemple suivant met à jour les paramètres de votre lac de données Amazon Security Lake. Vous pouvez utiliser cette opération pour spécifier le chiffrement des données, le stockage et les paramètres de région cumulative.

```
aws securitylake update-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

Sortie :


```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        },
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
```

```
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Getting started with Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

Exemple 2 : pour configurer votre lac de données dans une seule région

L'`create-data-lake`exemple suivant active Amazon Security Lake dans une seule AWS région et configure votre lac de données.

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
```

```
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

Sortie :


```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 500
                },
                "transitions": [
                    {
                        "days": 30,
                        "storageClass": "GLACIER"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
                "status": "INITIALIZED"
            }
        }
```

```
    ]
}
```

Pour plus d'informations, consultez Getting started with Amazon Security Lake dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous UpdateDataLakeà la section Référence des AWS CLI commandes.

`update-subscriber-notification`

L'exemple de code suivant montre comment utiliser`update-subscriber-notification`.

AWS CLI

Pour mettre à jour une notification d'abonné

L'`update-subscriber-notification`exemple suivant montre comment mettre à jour la méthode de notification pour un abonné.

```
aws securitylake update-subscriber-notification \
    --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
    --configuration '{"httpsNotificationConfiguration":
 {"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",
 "endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

Sortie :

```
{
    "subscriberEndpoint": [
        "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
    ]
}
```

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous UpdateSubscriberNotificationà la section Référence des AWS CLI commandes.

`update-subscriber`

L'exemple de code suivant montre comment utiliser`update-subscriber`.

AWS CLI

Pour mettre à jour un abonné Amazon Security Lake.

L'`update-subscriber`exemple suivant met à jour les sources d'accès aux données de Security Lake pour un abonné Security Lake spécifique.

```
aws securitylake update-subscriber \
    --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-04-19T15:19:44.421803+00:00",
        "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "LAMBDA_EXECUTION",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "EKS_AUDIT",
                    "sourceVersion": "2.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                }
```

```
            },
            {
                "awsLogSource": {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRXT0D4",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
                    },
                    "provider": {
                        "location": "amzn-s3-demo-bucket--usw2-az1--x-s3",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
                    },
                    "sourceName": "testCustom2"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "123456789012"
        },
        "subscriberName": "test",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-18T20:47:37.098000+00:00"
    }
```

```
}
```

Pour plus d'informations, consultez la section Gestion des abonnés dans le guide de l'utilisateur d'Amazon Security Lake.

- Pour plus de détails sur l'API, reportez-vous UpdateSubscriberà la section Référence des AWS CLI commandes.

AWS Serverless Application Repository exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS Serverless Application Repository.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`put-application-policy`

L'exemple de code suivant montre comment utiliser`put-application-policy`.

AWS CLI

Exemple 1 : pour partager une application publiquement

Ce qui suit `put-application-policy` partage une application publiquement, afin que tout le monde puisse la trouver et la déployer dans le AWS Serverless Application Repository.

```
aws serverlessrepo put-application-policy \
    --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
```

```
    --statements Principals='*',Actions=Deploy
```

Sortie :

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                ""
            ],
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

Exemple 2 : Pour partager une application en privé

Ce qui suit `put-application-policy` partage une application en privé, de sorte que seuls des AWS comptes spécifiques peuvent trouver et déployer votre application dans le AWS Serverless Application Repository.

```
aws serverlessrepo put-application-policy \
    --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
    --statements Principals=111111111111,222222222222,Actions=Deploy
```

Sortie :

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                "111111111111",
                "222222222222"
            ],
```

```
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Partage d'une application via la console dans le manuel AWS Serverless Application Repository Developer Guide

- Pour plus de détails sur l'API, reportez-vous PutApplicationPolicyà la section Référence des AWS CLI commandes.

Exemples de Service Catalog utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with Service Catalog.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`accept-portfolio-share`

L'exemple de code suivant montre comment utiliser`accept-portfolio-share`.

AWS CLI

Pour accepter une part de portefeuille

L'`accept-portfolio-share`exemple suivant accepte une offre, faite par un autre utilisateur, pour partager le portefeuille spécifié.

```
aws servicecatalog accept-portfolio-share \
```

```
    --portfolio-id port-2s6wuabcdefghijk
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AcceptPortfolioShareà la section Référence des AWS CLI commandes.

associate-principal-with-portfolio

L'exemple de code suivant montre comment utiliserassociate-principal-with-portfolio.

AWS CLI

Pour associer un mandant à un portefeuille

L'associate-principal-with-portfolioexemple suivant associe un utilisateur au portefeuille spécifié.

```
aws servicecatalog associate-principal-with-portfolio \
    --portfolio-id port-2s6abcdefwdh4 \
    --principal-arn arn:aws:iam::123456789012:user/usertest \
    --principal-type IAM
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AssociatePrincipalWithPortfolioà la section Référence des AWS CLI commandes.

associate-product-with-portfolio

L'exemple de code suivant montre comment utiliserassociate-product-with-portfolio.

AWS CLI

Pour associer un produit à un portefeuille

L'associate-product-with-portfolioexemple suivant associe le produit donné au portefeuille spécifié.

```
aws servicecatalog associate-product-with-portfolio
     --product-id prod-3p5abcdef3oyk
```

```
    --portfolio-id port-2s6abcdef5wdh4
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AssociateProductWithPortfolioà la section Référence des AWS CLI commandes.

`associate-tag-option-with-resource`

L'exemple de code suivant montre comment utiliser`associate-tag-option-with-resource`.

AWS CLI

Pour associer un TagOption à une ressource

L'`associate-tag-option-with-resource`exemple suivant associe la ressource spécifiée TagOption à la ressource spécifiée.

```
aws servicecatalog associate-tag-option-with-resource \
    --resource-id port-2s6abcdq5wdh4 \
    --tag-option-id tag-p3abc2pkpz5qc
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AssociateTagOptionWithResourceà la section Référence des AWS CLI commandes.

`copy-product`

L'exemple de code suivant montre comment utiliser`copy-product`.

AWS CLI

Pour copier un produit

L'`copy-product`exemple suivant crée une copie du produit spécifié en utilisant un fichier JSON pour transmettre les paramètres.

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

Contenu de `copy-product-input.json` :

```
{
    "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-
tcabcd3syn2xy",
    "TargetProductName": "copy-of-myproduct",
    "CopyOptions": [
        "CopyTags"
    ]
}
```

Sortie :

```
{
    "CopyProductToken": "copyproduct-abc5defgjkdji"
}
```

- Pour plus de détails sur l'API, reportez-vous CopyProductà la section Référence des AWS CLI commandes.

`create-portfolio-share`

L'exemple de code suivant montre comment utiliser`create-portfolio-share`.

AWS CLI

Pour partager un portefeuille avec un compte

L'`create-portfolio-share`exemple suivant partage le portefeuille spécifié avec le compte spécifié.

```
aws servicecatalog create-portfolio-share \
    --portfolio-id port-2s6abcdef5wdh4 \
    --account-id 794123456789
```

Cette commande ne produit aucune sortie.

- Pour plus de détails sur l'API, reportez-vous CreatePortfolioShareà la section Référence des AWS CLI commandes.

`create-portfolio`

L'exemple de code suivant montre comment utiliser`create-portfolio`.

AWS CLI

Pour créer un portefeuille

L'`create-portfolio`exemple suivant crée un portefeuille.

```
aws servicecatalog create-portfolio  \
    --provider-name my-provider \
    --display-name my-portfolio
```

Sortie :

```
{
    "PortfolioDetail": {
        "ProviderName": "my-provider",
        "DisplayName": "my-portfolio",
        "CreatedTime": 1571337221.555,
        "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-2s6xmplq5wdh4",
        "Id": "port-2s6xmplq5wdh4"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePortfolioà la section Référence des AWS CLI commandes.

`create-product`

L'exemple de code suivant montre comment utiliser`create-product`.

AWS CLI

Pour créer un produit

L'`create-product`exemple suivant crée un produit en utilisant un fichier JSON pour transmettre des paramètres.

```
aws servicecatalog create-product \
    --cli-input-json file://create-product-input.json
```

Contenu de `create-product-input.json` :

```
{
    "AcceptLanguage": "en",
    "Name": "test-product",
    "Owner": "test-owner",
    "Description": "test-description",
    "Distributor": "test-distributor",
    "SupportDescription": "test-support",
    "SupportEmail": "test@amazon.com",
    "SupportUrl": "https://aws.amazon.com",
    "ProductType": "CLOUD_FORMATION_TEMPLATE",
    "Tags": [
        {
            "Key": "region",
            "Value": "us-east-1"
        }
    ],
    "ProvisioningArtifactParameters": {
        "Name": "test-version-name",
        "Description": "test-version-description",
        "Info": {
            "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
        },
        "Type": "CLOUD_FORMATION_TEMPLATE"
    }
}
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "region",
            "Value": "us-east-1"
        }
    ],
    "ProductViewDetail": {
        "CreatedTime": 1576025036.0,
        "ProductARN": "arn:aws:catalog:us-west-2:1234568542028:product/
prod-3p5abcdef3oyk",
        "Status": "CREATED",
        "ProductViewSummary": {
            "Type": "CLOUD_FORMATION_TEMPLATE",
```

```
            "Distributor": "test-distributor",
            "SupportUrl": "https://aws.amazon.com",
            "SupportEmail": "test@amazon.com",
            "Id": "prodview-abcd42wvx45um",
            "SupportDescription": "test-support",
            "ShortDescription": "test-description",
            "Owner": "test-owner",
            "Name": "test-product2",
            "HasDefaultPath": false,
            "ProductId": "prod-3p5abcdef3oyk"
        }
    },
    "ProvisioningArtifactDetail": {
        "CreatedTime": 1576025036.0,
        "Active": true,
        "Id": "pa-pq3p5lil12a34",
        "Description": "test-version-description",
        "Name": "test-version-name",
        "Type": "CLOUD_FORMATION_TEMPLATE"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateProductà la section Référence des AWS CLI commandes.

`create-provisioning-artifact`

L'exemple de code suivant montre comment utiliser`create-provisioning-artifact`.

AWS CLI

Pour créer un artefact d'approvisionnement

L'`create-provisioning-artifact`exemple suivant crée un artefact de provisionnement en utilisant un fichier JSON pour transmettre des paramètres.

```
aws servicecatalog create-provisioning-artifact \
   --cli-input-json file://create-provisioning-artifact-input.json
```

Contenu de `create-provisioning-artifact-input.json` :

```
{
```

```
    "ProductId": "prod-nfi2abcdefghi",
    "Parameters": {
        "Name": "test-provisioning-artifact",
        "Description": "test description",
        "Info": {
            "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
        },
        "Type": "CLOUD_FORMATION_TEMPLATE"
    }
}
```

Sortie :

```
{
    "Info": {
        "TemplateUrl": "https://s3-us-west-1.amazonaws.com/cloudformation-templates-
us-west-1/my-cfn-template.template"
    },
    "Status": "CREATING",
    "ProvisioningArtifactDetail": {
        "Id": "pa-bb4abcdefwnaio",
        "Name": "test-provisioning-artifact",
        "Description": "test description",
        "Active": true,
        "Type": "CLOUD_FORMATION_TEMPLATE",
        "CreatedTime": 1576022545.0
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateProvisioningArtifactà la section Référence des AWS CLI commandes.

`create-tag-option`

L'exemple de code suivant montre comment utiliser`create-tag-option`.

AWS CLI

Pour créer un TagOption

L'`create-tag-option`exemple suivant crée un TagOption.

```
aws servicecatalog create-tag-option
    --key 1234
    --value name
```

Sortie :

```
{
    "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateTagOptionà la section Référence des AWS CLI commandes.

`delete-portfolio-share`

L'exemple de code suivant montre comment utiliser`delete-portfolio-share`.

AWS CLI

Pour arrêter de partager un portefeuille avec un compte

L'`delete-portfolio-share`exemple suivant arrête de partager le portefeuille avec le compte spécifié.

```
aws servicecatalog delete-portfolio-share \
    --portfolio-id port-2s6abcdq5wdh4 \
    --account-id 123456789012
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePortfolioShareà la section Référence des AWS CLI commandes.

`delete-portfolio`

L'exemple de code suivant montre comment utiliser`delete-portfolio`.

AWS CLI

Pour supprimer un portefeuille

L'`delete-portfolio`exemple suivant supprime le portefeuille spécifié.

```
aws servicecatalog delete-portfolio \
    --id port-abcdlx4gox4do
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePortfolioà la section Référence des AWS CLI commandes.

`delete-product`

L'exemple de code suivant montre comment utiliser`delete-product`.

AWS CLI

Pour supprimer un produit

L'`delete-product`exemple suivant supprime le produit spécifié.

```
aws servicecatalog delete-product \
    --id prod-abcdcek6yhbxi
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteProductà la section Référence des AWS CLI commandes.

`delete-provisioning-artifact`

L'exemple de code suivant montre comment utiliser`delete-provisioning-artifact`.

AWS CLI

Pour supprimer un artefact d'approvisionnement

L'`delete-provisioning-artifact`exemple suivant supprime l'artefact de provisionnement spécifié.

```
aws servicecatalog delete-provisioning-artifact \
    --product-id prod-abc2uebuplcpw \
    --provisioning-artifact-id pa-pqabcddii7ouc
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteProvisioningArtifactà la section Référence des AWS CLI commandes.

`delete-tag-option`

L'exemple de code suivant montre comment utiliser`delete-tag-option`.

AWS CLI

Pour supprimer un TagOption

L'`delete-tag-option`exemple suivant supprime le paramètre spécifié TagOption.

```
aws servicecatalog delete-tag-option \
    --id tag-iabcdn4fzjjms
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteTagOptionà la section Référence des AWS CLI commandes.

`describe-copy-product-status`

L'exemple de code suivant montre comment utiliser`describe-copy-product-status`.

AWS CLI

Pour décrire l'état de l'opération de copie du produit

L'`describe-copy-product-status`exemple suivant affiche l'état actuel de l'opération de copie asynchrone spécifiée.

```
aws servicecatalog describe-copy-product-status \
    --copy-product-token copyproduct-znn5tf5abcd3w
```

Sortie :

```
{
    "CopyProductStatus": "SUCCEEDED",
    "TargetProductId": "prod-os6hog7abcdt2"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeCopyProductStatusà la section Référence des AWS CLI commandes.

`describe-portfolio`

L'exemple de code suivant montre comment utiliser`describe-portfolio`.

AWS CLI

Pour décrire un portefeuille

L'`describe-portfolio`exemple suivant affiche les détails du portefeuille spécifié.

```
aws servicecatalog describe-portfolio \
    --id port-2s6abcdq5wdh4
```

Sortie :

```
{
    "TagOptions": [],
    "PortfolioDetail": {
        "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/
port-2s6abcdq5wdh4",
        "Id": "port-2s6wuzyq5wdh4",
        "CreatedTime": 1571337221.555,
        "DisplayName": "my-portfolio",
        "ProviderName": "my-provider"
    },
    "Tags": []
}
```

- Pour plus de détails sur l'API, reportez-vous DescribePortfolioà la section Référence des AWS CLI commandes.

`describe-product-as-admin`

L'exemple de code suivant montre comment utiliser`describe-product-as-admin`.

AWS CLI

Pour décrire un produit en tant qu'administrateur

L'`describe-product-as-admin`exemple suivant affiche les détails du produit spécifié à l'aide des privilèges d'administrateur.

```
aws servicecatalog describe-product-as-admin \
    --id prod-abcdcek6yhbxi
```

Sortie :

```
{
    "TagOptions": [],
    "ProductViewDetail": {
        "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-
abcdcek6yhbxi",
        "ProductViewSummary": {
            "SupportEmail": "test@amazon.com",
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "Distributor": "test-distributor",
            "ShortDescription": "test-description",
            "Owner": "test-owner",
            "Id": "prodview-wi3l2j4abc6vc",
            "SupportDescription": "test-support",
            "ProductId": "prod-abcdcek6yhbxi",
            "HasDefaultPath": false,
            "Name": "test-product3",
            "SupportUrl": "https://aws.amazon.com"
        },
        "CreatedTime": 1577136715.0,
        "Status": "CREATED"
    },
    "ProvisioningArtifactSummaries": [
        {
            "CreatedTime": 1577136715.0,
            "Description": "test-version-description",
            "ProvisioningArtifactMetadata": {
                "SourceProvisioningArtifactId": "pa-abcdxkkiv5fcm"
```

```
            },
            "Name": "test-version-name-3",
            "Id": "pa-abcdxkkiv5fcm"
        }
    ],
    "Tags": [
        {
            "Value": "iad",
            "Key": "region"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeProductAsAdminà la section Référence des AWS CLI commandes.

`describe-provisioned-product`

L'exemple de code suivant montre comment utiliser`describe-provisioned-product`.

AWS CLI

Pour décrire un produit approvisionné

L'`describe-provisioned-product`exemple suivant affiche les détails du produit approvisionné spécifié.

```
aws servicecatalog describe-provisioned-product \
    --id pp-dpom27bm4abcd
```

Sortie :

```
{
    "ProvisionedProductDetail": {
        "Status": "ERROR",
        "CreatedTime": 1577222793.358,
        "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/mytestppname3/
pp-dpom27bm4abcd",
        "Id": "pp-dpom27bm4abcd",
        "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
```

```
        "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
        "LastRecordId": "rec-tfuawdjovzxge",
        "Type": "CFN_STACK",
        "Name": "mytestppname3"
    },
    "CloudWatchDashboards": []
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeProvisionedProductà la section Référence des AWS CLI commandes.

`describe-provisioning-artifact`

L'exemple de code suivant montre comment utiliser`describe-provisioning-artifact`.

AWS CLI

Pour décrire un artefact d'approvisionnement

L'`describe-provisioning-artifact`exemple suivant affiche les détails de l'artefact de provisionnement spécifié.

```
aws servicecatalog describe-provisioning-artifact \
    --provisioning-artifact-id pa-pcz347abcdcfm \
    --product-id prod-abcdfz3syn2rg
```

Sortie :

```
{
    "Info": {
        "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
    },
    "ProvisioningArtifactDetail": {
        "Id": "pa-pcz347abcdcfm",
        "Active": true,
        "Type": "CLOUD_FORMATION_TEMPLATE",
        "Description": "updated description",
        "CreatedTime": 1562097906.0,
        "Name": "updated name"
    },
    "Status": "AVAILABLE"
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeProvisioningArtifactà la section Référence des AWS CLI commandes.

`describe-tag-option`

L'exemple de code suivant montre comment utiliser`describe-tag-option`.

AWS CLI

Pour décrire un TagOption

L'`describe-tag-option`exemple suivant affiche les détails de ce qui est spécifié TagOption.

```
aws servicecatalog describe-tag-option \
    --id tag-p3tej2abcd5qc
```

Sortie :

```
{
    "TagOptionDetail": {
        "Active": true,
        "Id": "tag-p3tej2abcd5qc",
        "Value": "value-3",
        "Key": "1234"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTagOptionà la section Référence des AWS CLI commandes.

`disassociate-principal-from-portfolio`

L'exemple de code suivant montre comment utiliser`disassociate-principal-from-portfolio`.

AWS CLI

Pour dissocier un capital d'un portefeuille

L'`disassociate-principal-from-portfolio`exemple suivant dissocie le capital spécifié du portefeuille.

```
aws servicecatalog disassociate-principal-from-portfolio \
    --portfolio-id port-2s6abcdq5wdh4 \
    --principal-arn arn:aws:iam::123456789012:group/myendusers
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DisassociatePrincipalFromPortfolioà la section Référence des AWS CLI commandes.

`disassociate-product-from-portfolio`

L'exemple de code suivant montre comment utiliser`disassociate-product-from-portfolio`.

AWS CLI

Pour dissocier un produit d'un portefeuille

L'`disassociate-product-from-portfolio`exemple suivant dissocie le produit spécifié du portefeuille.

```
aws servicecatalog disassociate-product-from-portfolio \
    --product-id prod-3p5abcdmu3oyk \
    --portfolio-id port-2s6abcdq5wdh4
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DisassociateProductFromPortfolioà la section Référence des AWS CLI commandes.

`disassociate-tag-option-from-resource`

L'exemple de code suivant montre comment utiliser`disassociate-tag-option-from-resource`.

AWS CLI

Pour dissocier une TagOption ressource

L'`disassociate-tag-option-from-resource`exemple suivant dissocie le spécifié `TagOption` de la ressource.

```
aws servicecatalog disassociate-tag-option-from-resource \
     --resource-id port-2s6abcdq5wdh4 \
     --tag-option-id tag-p3abc2pkpz5qc
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DisassociateTagOptionFromResourceà la section Référence des AWS CLI commandes.

`list-accepted-portfolio-shares`

L'exemple de code suivant montre comment utiliser`list-accepted-portfolio-shares`.

AWS CLI

Pour répertorier les actions de portefeuille acceptées

L'`list-accepted-portfolio-shares`exemple suivant répertorie tous les portefeuilles dont le partage a été accepté par ce compte, y compris uniquement les portefeuilles Service Catalog par défaut.

```
aws servicecatalog list-accepted-portfolio-shares \
    --portfolio-share-type "AWS_SERVICECATALOG"
```

Sortie :

```
{
    "PortfolioDetails": [
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-
d2abcd5dpkuma",
            "Description": "AWS Service Catalog Reference blueprints for often-used
 AWS services such as EC2, S3, RDS, VPC and EMR.",
            "CreatedTime": 1574456190.687,
            "ProviderName": "AWS Service Catalog",
            "DisplayName": "Reference Architectures",
            "Id": "port-d2abcd5dpkuma"
        },
```

```
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-
abcdefaua7zpu",
            "Description": "AWS well-architected blueprints for high reliability
 applications.",
            "CreatedTime": 1574461496.092,
            "ProviderName": "AWS Service Catalog",
            "DisplayName": "High Reliability Architectures",
            "Id": "port-abcdefaua7zpu"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAcceptedPortfolioSharesà la section Référence des AWS CLI commandes.

`list-portfolio-access`

L'exemple de code suivant montre comment utiliser`list-portfolio-access`.

AWS CLI

Pour répertorier les comptes ayant accès à un portefeuille

L'`list-portfolio-access`exemple suivant répertorie les AWS comptes qui ont accès au portefeuille spécifié.

```
aws servicecatalog list-portfolio-access \
    --portfolio-id port-2s6abcdq5wdh4
```

Sortie :

```
{
    "AccountIds": [
        "123456789012"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPortfolioAccessà la section Référence des AWS CLI commandes.

`list-portfolios-for-product`

L'exemple de code suivant montre comment utiliser`list-portfolios-for-product`.

AWS CLI

Pour répertorier les portefeuilles associés à un produit

L'`list-portfolios-for-product`exemple suivant répertorie les portefeuilles associés au produit spécifié.

```
aws servicecatalog list-portfolios-for-product \
    --product-id prod-abcdfz3syn2rg
```

Sortie :

```
{
    "PortfolioDetails": [
        {
            "CreatedTime": 1571337221.555,
            "Id": "port-2s6abcdq5wdh4",
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
            "DisplayName": "my-portfolio",
            "ProviderName": "my-provider"
        },
        {
            "CreatedTime": 1559665256.348,
            "Id": "port-5abcd3e5st4ei",
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-5abcd3e5st4ei",
            "DisplayName": "test",
            "ProviderName": "provider-name"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPortfoliosForProductà la section Référence des AWS CLI commandes.

`list-portfolios`

L'exemple de code suivant montre comment utiliser`list-portfolios`.

AWS CLI

Pour répertorier les portefeuilles

L'`list-portfolios`exemple suivant répertorie les portefeuilles Service Catalog de la région actuelle.

```
aws servicecatalog list-portfolios
```

Sortie :

```
{
    "PortfolioDetails": [
        {
           "CreatedTime": 1559665256.348,
           "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmplst4ei",
           "DisplayName": "my-portfolio",
           "Id": "port-5pzcxmplst4ei",
           "ProviderName": "my-user"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPortfoliosà la section Référence des AWS CLI commandes.

`list-principals-for-portfolio`

L'exemple de code suivant montre comment utiliser`list-principals-for-portfolio`.

AWS CLI

Pour répertorier tous les principaux d'un portefeuille

L'`list-principals-for-portfolio`exemple suivant répertorie tous les principes du portefeuille spécifié.

```
aws servicecatalog list-principals-for-portfolio \
    --portfolio-id port-2s6abcdq5wdh4
```

Sortie :

```
{
    "Principals": [
        {
            "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
            "PrincipalType": "IAM"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPrincipalsForPortfolioà la section Référence des AWS CLI commandes.

`list-provisioning-artifacts`

L'exemple de code suivant montre comment utiliser`list-provisioning-artifacts`.

AWS CLI

Pour répertorier tous les artefacts d'approvisionnement d'un produit

L'`list-provisioning-artifacts`exemple suivant répertorie tous les artefacts d'approvisionnement pour le produit spécifié.

```
aws servicecatalog list-provisioning-artifacts \
    --product-id prod-nfi2abcdefgcpw
```

Sortie :

```
{
    "ProvisioningArtifactDetails": [
        {
            "Id": "pa-abcdef54ipm6z",
            "Description": "test-version-description",
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "CreatedTime": 1576021147.0,
            "Active": true,
```

```
            "Name": "test-version-name"
        },
        {
            "Id": "pa-bb4zyxwwnaio",
            "Description": "test description",
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "CreatedTime": 1576022545.0,
            "Active": true,
            "Name": "test-provisioning-artifact-2"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListProvisioningArtifactsà la section Référence des AWS CLI commandes.

`list-resources-for-tag-option`

L'exemple de code suivant montre comment utiliser`list-resources-for-tag-option`.

AWS CLI

Pour répertorier les ressources associées à un TagOption

L'`list-resources-for-tag-option`exemple suivant répertorie les ressources associées aux informations spécifiées`TagOption`.

```
aws servicecatalog list-resources-for-tag-option \
    --tag-option-id tag-p3tej2abcd5qc
```

Sortie :

```
{
    "ResourceDetails": [
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
            "Name": "my product",
            "Description": "description",
            "CreatedTime": 1562097906.0,
            "Id": "prod-abcdfz3syn2rg"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourcesForTagOptionà la section Référence des AWS CLI commandes.

`list-tag-options`

L'exemple de code suivant montre comment utiliser`list-tag-options`.

AWS CLI

L'`list-tag-options`exemple suivant répertorie toutes les valeurs de`TagOptions`.

```
aws servicecatalog list-tag-options
```

Sortie :

```
{
    "TagOptionDetails": [
        {
            "Value": "newvalue",
            "Active": true,
            "Id": "tag-iabcdn4fzjjms",
            "Key": "1234"
        },
        {
            "Value": "value1",
            "Active": true,
            "Id": "tag-e3abcdvmwvrzy",
            "Key": "key"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagOptionsà la section Référence des AWS CLI commandes.

`provision-product`

L'exemple de code suivant montre comment utiliser`provision-product`.

AWS CLI

Pour approvisionner un produit

L'`provision-product`exemple suivant approvisionne le produit spécifié à l'aide de l'artefact d'approvisionnement spécifié.

```
aws servicecatalog provision-product \
    --product-id prod-abcdfz3syn2rg \
    --provisioning-artifact-id pa-abc347pcsccfm \
    --provisioned-product-name "mytestppname3"
```

Sortie :

```
{
    "RecordDetail": {
        "RecordId": "rec-tfuawdabcdege",
        "CreatedTime": 1577222793.362,
        "ProvisionedProductId": "pp-abcd27bm4mldq",
        "PathId": "lpv2-abcdg3jp6t5k6",
        "RecordErrors": [],
        "ProductId": "prod-abcdfz3syn2rg",
        "UpdatedTime": 1577222793.362,
        "RecordType": "PROVISION_PRODUCT",
        "ProvisionedProductName": "mytestppname3",
        "ProvisioningArtifactId": "pa-pcz347abcdcfm",
        "RecordTags": [],
        "Status": "CREATED",
        "ProvisionedProductType": "CFN_STACK"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ProvisionProductà la section Référence des AWS CLI commandes.

`reject-portfolio-share`

L'exemple de code suivant montre comment utiliser`reject-portfolio-share`.

AWS CLI

Pour rejeter une action de portefeuille

L'`reject-portfolio-share`exemple suivant rejette la part du portefeuille pour le portefeuille donné.

```
aws servicecatalog reject-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RejectPortfolioShareà la section Référence des AWS CLI commandes.

`scan-provisioned-products`

L'exemple de code suivant montre comment utiliser`scan-provisioned-products`.

AWS CLI

Pour répertorier tous les produits approvisionnés disponibles

L'`scan-provisioned-products`exemple suivant répertorie les produits provisionnés disponibles.

```
aws servicecatalog scan-provisioned-products
```

Sortie :

```
{
    "ProvisionedProducts": [
        {
            "Status": "ERROR",
            "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
            "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
            "Id": "pp-abcd27bm4mldq",
            "Type": "CFN_STACK",
            "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
            "CreatedTime": 1577222793.358,
            "Name": "mytestppname3",
            "LastRecordId": "rec-tfuawdabcdxge"
        }
```

```
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ScanProvisionedProductsà la section Référence des AWS CLI commandes.

`search-products-as-admin`

L'exemple de code suivant montre comment utiliser`search-products-as-admin`.

AWS CLI

Pour rechercher des produits dotés de privilèges d'administrateur

L'`search-products-as-admin`exemple suivant recherche des produits dotés de privilèges d'administrateur, en utilisant un identifiant de portefeuille comme filtre.

```
aws servicecatalog search-products-as-admin \
    --portfolio-id port-5abcd3e5st4ei
```

Sortie :

```
{
    "ProductViewDetails": [
        {
            "ProductViewSummary": {
                "Name": "my product",
                "Owner": "owner name",
                "Type": "CLOUD_FORMATION_TEMPLATE",
                "ProductId": "prod-abcdfz3syn2rg",
                "HasDefaultPath": false,
                "Id": "prodview-abcdmyuzv2dlu",
                "ShortDescription": "description"
            },
            "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
            "CreatedTime": 1562097906.0,
            "Status": "CREATED"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous SearchProductsAsAdminà la section Référence des AWS CLI commandes.

`search-provisioned-products`

L'exemple de code suivant montre comment utiliser`search-provisioned-products`.

AWS CLI

Pour rechercher des produits approvisionnés

L'`search-provisioned-products`exemple suivant recherche les produits approvisionnés correspondant à l'ID de produit spécifié, à l'aide d'un fichier JSON pour transmettre les paramètres.

```
aws servicecatalog search-provisioned-products \
    --cli-input-json file://search-provisioned-products-input.json
```

Contenu de `search-provisioned-products-input.json` :

```
{
    "Filters": {
        "SearchQuery": [
            "prod-tcjvfz3syn2rg"
        ]
    }
}
```

Sortie :

```
{
    "ProvisionedProducts": [
        {
            "ProvisioningArtifactId": "pa-pcz347abcdcfm",
            "Name": "mytestppname3",
            "CreatedTime": 1577222793.358,
            "Id": "pp-abcd27bm4mldq",
            "Status": "ERROR",
            "UserArn": "arn:aws:iam::123456789012:user/cliuser",
            "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
```

```
            "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
            "Tags": [
                {
                    "Value": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
                    "Key": "aws:servicecatalog:productArn"
                },
                {
                    "Value": "arn:aws:iam::123456789012:user/cliuser",
                    "Key": "aws:servicecatalog:provisioningPrincipalArn"
                },
                {
                    "Value": "value-3",
                    "Key": "1234"
                },
                {
                    "Value": "pa-pcz347abcdcfm",
                    "Key": "aws:servicecatalog:provisioningArtifactIdentifier"
                },
                {
                    "Value": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
                    "Key": "aws:servicecatalog:portfolioArn"
                },
                {
                    "Value": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
                    "Key": "aws:servicecatalog:provisionedProductArn"
                }
            ],
            "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
            "UserArnSession": "arn:aws:iam::123456789012:user/cliuser",
            "Type": "CFN_STACK",
            "LastRecordId": "rec-tfuawdabcdxge",
            "ProductId": "prod-abcdfz3syn2rg"
        }
    ],
    "TotalResultsCount": 1
}
```

- Pour plus de détails sur l'API, reportez-vous SearchProvisionedProductsà la section Référence des AWS CLI commandes.

update-portfolio

L'exemple de code suivant montre comment utiliserupdate-portfolio.

AWS CLI

Pour mettre à jour un portefeuille

L'update-portfolioexemple suivant met à jour le nom du portefeuille spécifié.

```
aws servicecatalog update-portfolio \
    --id port-5abcd3e5st4ei \
    --display-name "New portfolio name"
```

Sortie :

```
{
    "PortfolioDetail": {
        "DisplayName": "New portfolio name",
        "ProviderName": "provider",
        "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-5abcd3e5st4ei",
        "Id": "port-5abcd3e5st4ei",
        "CreatedTime": 1559665256.348
    },
    "Tags": []
}
```

- Pour plus de détails sur l'API, reportez-vous UpdatePortfolioà la section Référence des AWS CLI commandes.

update-product

L'exemple de code suivant montre comment utiliserupdate-product.

AWS CLI

Pour mettre à jour un produit

L'update-productexemple suivant met à jour le nom et le propriétaire du produit spécifié.

```
aws servicecatalog update-product \
```

```
    --id prod-os6abc7drqlt2 \
    --name "New product name" \
    --owner "Updated product owner"
```

Sortie :

```
{
    "Tags": [
        {
            "Value": "iad",
            "Key": "region"
        }
    ],
    "ProductViewDetail": {
        "ProductViewSummary": {
            "Owner": "Updated product owner",
            "ProductId": "prod-os6abc7drqlt2",
            "Distributor": "test-distributor",
            "SupportUrl": "https://aws.amazon.com",
            "Name": "New product name",
            "ShortDescription": "test-description",
            "HasDefaultPath": false,
            "Id": "prodview-6abcdgrfhvidy",
            "SupportDescription": "test-support",
            "SupportEmail": "test@amazon.com",
            "Type": "CLOUD_FORMATION_TEMPLATE"
        },
        "Status": "CREATED",
        "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
os6abc7drqlt2",
        "CreatedTime": 1577136255.0
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateProductà la section Référence des AWS CLI commandes.

`update-provisioning-artifact`

L'exemple de code suivant montre comment utiliser`update-provisioning-artifact`.

AWS CLI

Pour mettre à jour un artefact d'approvisionnement

L'`update-provisioning-artifact`exemple suivant met à jour le nom et la description de l'artefact de provisionnement spécifié, en utilisant un fichier JSON pour transmettre les paramètres.

```
aws servicecatalog update-provisioning-artifact \
    --cli-input-json file://update-provisioning-artifact-input.json
```

Contenu de `update-provisioning-artifact-input.json` :

```
{
    "ProductId": "prod-abcdfz3syn2rg",
    "ProvisioningArtifactId": "pa-pcz347abcdcfm",
    "Name": "updated name",
    "Description": "updated description"
}
```

Sortie :

```
{
    "Info": {
        "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
    },
    "Status": "AVAILABLE",
    "ProvisioningArtifactDetail": {
        "Active": true,
        "Description": "updated description",
        "Id": "pa-pcz347abcdcfm",
        "Name": "updated name",
        "Type": "CLOUD_FORMATION_TEMPLATE",
        "CreatedTime": 1562097906.0
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateProvisioningArtifactà la section Référence des AWS CLI commandes.

`update-tag-option`

L'exemple de code suivant montre comment utiliser`update-tag-option`.

AWS CLI

Pour mettre à jour un TagOption

L'`update-tag-option`exemple suivant met à jour la valeur de a `TagOption` en utilisant le fichier JSON spécifié.

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-input.json
```

Contenu de `update-tag-option-input.json` :

```
{
    "Id": "tag-iabcdn4fzjjms",
    "Value": "newvalue",
    "Active": true
}
```

Sortie :

```
{
    "TagOptionDetail": {
        "Value": "newvalue",
        "Key": "1234",
        "Active": true,
        "Id": "tag-iabcdn4fzjjms"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateTagOptionà la section Référence des AWS CLI commandes.

Exemples de quotas de service utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants en utilisant le service AWS Command Line Interface with Service Quotas.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-aws-default-service-quota`

L'exemple de code suivant montre comment utiliser`get-aws-default-service-quota`.

AWS CLI

Pour décrire un quota de service par défaut

L'`get-aws-default-service-quota`exemple suivant affiche les détails du quota spécifié.

```
aws service-quotas get-aws-default-service-quota \
    --service-code ec2 \
    --quota-code L-1216C47A
```

Sortie :

```
{
    "Quota": {
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaArn": "arn:aws:servicequotas:us-east-2::ec2/L-1216C47A",
        "QuotaCode": "L-1216C47A",
        "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)
 instances",
        "Value": 5.0,
        "Unit": "None",
        "Adjustable": true,
        "GlobalQuota": false,
        "UsageMetric": {
```

```
            "MetricNamespace": "AWS/Usage",
            "MetricName": "ResourceCount",
            "MetricDimensions": {
                "Class": "Standard/OnDemand",
                "Resource": "vCPU",
                "Service": "EC2",
                "Type": "Resource"
            },
            "MetricStatisticRecommendation": "Maximum"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetAwsDefaultServiceQuotaà la section Référence des AWS CLI commandes.

`get-requested-service-quota-change`

L'exemple de code suivant montre comment utiliser`get-requested-service-quota-change`.

AWS CLI

Pour décrire une demande d'augmentation de quota de service

L'`get-requested-service-quota-change`exemple suivant décrit la demande d'augmentation de quota spécifiée.

```
aws service-quotas get-requested-service-quota-change \
    --request-id d187537d15254312a9609aa51bbf7624u7W49tPO
```

Sortie :

```
{
    "RequestedQuota": {
        "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
        "CaseId": "6780195351",
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaCode": "L-20F13EBD",
        "QuotaName": "Running Dedicated c5n Hosts",
        "DesiredValue": 2.0,
        "Status": "CASE_OPENED",
```

```
        "Created": 1580446904.067,
        "LastUpdated": 1580446953.265,
        "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
        "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",
        "GlobalQuota": false,
        "Unit": "None"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetRequestedServiceQuotaChangeà la section Référence des AWS CLI commandes.

`get-service-quota`

L'exemple de code suivant montre comment utiliser`get-service-quota`.

AWS CLI

Pour décrire un quota de service

L'`get-service-quota`exemple suivant affiche les détails relatifs au quota spécifié.

```
aws service-quotas get-service-quota \
    --service-code ec2 \
    --quota-code L-1216C47A
```

Sortie :

```
{
    "Quota": {
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-1216C47A",
        "QuotaCode": "L-1216C47A",
        "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)
 instances",
        "Value": 1920.0,
        "Unit": "None",
        "Adjustable": true,
        "GlobalQuota": false,
        "UsageMetric": {
```

```
            "MetricNamespace": "AWS/Usage",
            "MetricName": "ResourceCount",
            "MetricDimensions": {
                "Class": "Standard/OnDemand",
                "Resource": "vCPU",
                "Service": "EC2",
                "Type": "Resource"
            },
            "MetricStatisticRecommendation": "Maximum"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetServiceQuotaà la section Référence des AWS CLI commandes.

`list-aws-default-service-quotas`

L'exemple de code suivant montre comment utiliser`list-aws-default-service-quotas`.

AWS CLI

Pour répertorier les quotas par défaut pour un service

L'`list-aws-default-service-quotas`exemple suivant répertorie les valeurs par défaut des quotas pour le service spécifié.

```
aws service-quotas list-aws-default-service-quotas \
    --service-code xray
```

Sortie :

```
{
    "Quotas": [
        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray",
            "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-C6B6F05D",
            "QuotaCode": "L-C6B6F05D",
            "QuotaName": "Indexed annotations per trace",
            "Value": 50.0,
            "Unit": "None",
```

```
            "Adjustable": false,
            "GlobalQuota": false
        },
        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray",
            "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-D781C0FD",
            "QuotaCode": "L-D781C0FD",
            "QuotaName": "Segment document size",
            "Value": 64.0,
            "Unit": "Kilobytes",
            "Adjustable": false,
            "GlobalQuota": false
        },
        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray",
            "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-998BFF16",
            "QuotaCode": "L-998BFF16",
            "QuotaName": "Trace and service graph retention in days",
            "Value": 30.0,
            "Unit": "None",
            "Adjustable": false,
            "GlobalQuota": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAwsDefaultServiceQuotasà la section Référence des AWS CLI commandes.

`list-requested-service-quota-change-history-by-quota`

L'exemple de code suivant montre comment utiliser`list-requested-service-quota-change-history-by-quota`.

AWS CLI

Pour répertorier vos demandes d'augmentation de quota

L'`list-requested-service-quota-change-history-by-quota`exemple suivant répertorie les demandes d'augmentation de quota pour le quota spécifié.

```
aws service-quotas list-requested-service-quota-change-history-by-quota \
    --service-code ec2 \
    --quota-code L-20F13EBD
```

Sortie :

```
{
    "RequestedQuotas": [
        {
            "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
            "CaseId": "6780195351",
            "ServiceCode": "ec2",
            "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
            "QuotaCode": "L-20F13EBD",
            "QuotaName": "Running Dedicated c5n Hosts",
            "DesiredValue": 2.0,
            "Status": "CASE_OPENED",
            "Created": 1580446904.067,
            "LastUpdated": 1580446953.265,
            "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
            "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
            "GlobalQuota": false,
            "Unit": "None"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListRequestedServiceQuotaChangeHistoryByQuotaà la section Référence des AWS CLI commandes.

`list-requested-service-quota-change-history`

L'exemple de code suivant montre comment utiliser`list-requested-service-quota-change-history`.

AWS CLI

Pour répertorier vos demandes d'augmentation de quota

L'`list-requested-service-quota-change-history`exemple suivant répertorie les demandes d'augmentation de quota pour le service spécifié.

```
aws service-quotas list-requested-service-quota-change-history \
    --service-code ec2
```

Sortie :

```
{
    "RequestedQuotas": [
        {
            "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
            "CaseId": "6780195351",
            "ServiceCode": "ec2",
            "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
            "QuotaCode": "L-20F13EBD",
            "QuotaName": "Running Dedicated c5n Hosts",
            "DesiredValue": 2.0,
            "Status": "CASE_OPENED",
            "Created": 1580446904.067,
            "LastUpdated": 1580446953.265,
            "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
            "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
            "GlobalQuota": false,
            "Unit": "None"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListRequestedServiceQuotaChangeHistoryà la section Référence des AWS CLI commandes.

`list-service-quotas`

L'exemple de code suivant montre comment utiliser`list-service-quotas`.

AWS CLI

Pour répertorier les quotas d'un service

L'`list-service-quotas`exemple suivant affiche des détails sur les quotas pour AWS CloudFormation.

```
aws service-quotas list-service-quotas \
    --service-code cloudformation
```

Sortie :

```
{
    "Quotas": [
        {
            "ServiceCode": "cloudformation",
            "ServiceName": "AWS CloudFormation",
            "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-87D14FB7",
            "QuotaCode": "L-87D14FB7",
            "QuotaName": "Output count in CloudFormation template",
            "Value": 60.0,
            "Unit": "None",
            "Adjustable": false,
            "GlobalQuota": false
        },
        {
            "ServiceCode": "cloudformation",
            "ServiceName": "AWS CloudFormation",
            "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-0485CB21",
            "QuotaCode": "L-0485CB21",
            "QuotaName": "Stack count",
            "Value": 200.0,
            "Unit": "None",
            "Adjustable": true,
            "GlobalQuota": false
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListServiceQuotasà la section Référence des AWS CLI commandes.

`list-services`

L'exemple de code suivant montre comment utiliser`list-services`.

AWS CLI

Pour répertorier les services disponibles

La commande suivante répertorie les services disponibles dans Service Quotas.

```
aws service-quotas list-services
```

Sortie :

```
{
    "Services": [
        {
            "ServiceCode": "AWSCloudMap",
            "ServiceName": "AWS Cloud Map"
        },
        {
            "ServiceCode": "access-analyzer",
            "ServiceName": "Access Analyzer"
        },
        {
            "ServiceCode": "acm",
            "ServiceName": "AWS Certificate Manager (ACM)"
        },

        ...truncated...

        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray"
        }
    ]
}
```

Vous pouvez ajouter le `--query` paramètre pour filtrer l'affichage en fonction des informations qui vous intéressent. L'exemple suivant affiche uniquement les codes de service.

```
aws service-quotas list-services \
```

```
    --query Services[*].ServiceCode
```

Sortie :

```
[
    "AWSCloudMap",
    "access-analyzer",
    "acm",
    "acm-pca",
    "amplify",
    "apigateway",
    "application-autoscaling",
        ...truncated...
    "xray"
]
```

- Pour plus de détails sur l'API, reportez-vous ListServicesà la section Référence des AWS CLI commandes.

`request-service-quota-increase`

L'exemple de code suivant montre comment utiliser`request-service-quota-increase`.

AWS CLI

Pour demander une augmentation du quota de service

L'`request-service-quota-increase`exemple suivant demande une augmentation du quota de service spécifié.

```
aws service-quotas request-service-quota-increase \
    --service-code ec2 \
    --quota-code L-20F13EBD \
    --desired-value 2
```

Sortie :

```
{
    "RequestedQuota": {
        "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
        "ServiceCode": "ec2",
```

```
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaCode": "L-20F13EBD",
        "QuotaName": "Running Dedicated c5n Hosts",
        "DesiredValue": 2.0,
        "Status": "PENDING",
        "Created": 1580446904.067,
        "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
        "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",
        "GlobalQuota": false,
        "Unit": "None"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous RequestServiceQuotaIncreaseà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon SES AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon SES.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`delete-identity`

L'exemple de code suivant montre comment utiliser`delete-identity`.

AWS CLI

Pour supprimer une identité

L'exemple suivant utilise la commande `delete-identity` pour supprimer une identité de la liste des identités vérifiées auprès d'Amazon SES :

```
aws ses delete-identity --identity user@example.com
```

Pour plus d'informations sur les identités vérifiées, consultez Vérification des adresses e-mail et des domaines dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous DeleteIdentityà la section Référence des AWS CLI commandes.

`get-identity-dkim-attributes`

L'exemple de code suivant montre comment utiliser`get-identity-dkim-attributes`.

AWS CLI

Pour obtenir les attributs Amazon SES Easy DKIM pour une liste d'identités

L'exemple suivant utilise la `get-identity-dkim-attributes` commande pour récupérer les attributs Amazon SES Easy DKIM pour une liste d'identités :

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

Sortie :

```
{
   "DkimAttributes": {
       "example.com": {
           "DkimTokens": [
               "EXAMPLEjcs5xoyqytjsotsijas7236gr",
               "EXAMPLEjr76cvoc6mysspnioorxsn6ep",
               "EXAMPLEkbmkqkhlm2lyz77ppkulerm4k"
           ],
           "DkimEnabled": true,
           "DkimVerificationStatus": "Success"
       },
       "user@example.com": {
           "DkimEnabled": false,
           "DkimVerificationStatus": "NotStarted"
```

```
        }
    }
}
```

Si vous appelez cette commande avec une identité que vous n'avez jamais soumise pour vérification, cette identité n'apparaîtra pas dans la sortie.

Pour plus d'informations sur Easy DKIM, consultez Easy DKIM dans Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetIdentityDkimAttributesà la section Référence des AWS CLI commandes.

`get-identity-notification-attributes`

L'exemple de code suivant montre comment utiliser`get-identity-notification-attributes`.

AWS CLI

Pour obtenir les attributs de notification Amazon SES pour une liste d'identités

L'exemple suivant utilise la `get-identity-notification-attributes` commande pour récupérer les attributs de notification Amazon SES pour une liste d'identités :

```
aws ses get-identity-notification-attributes --
identities "user1@example.com" "user2@example.com"
```

Sortie :

```
{
    "NotificationAttributes": {
        "user1@example.com": {
            "ForwardingEnabled": false,
            "ComplaintTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
            "BounceTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
            "DeliveryTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic"
        },
        "user2@example.com": {
            "ForwardingEnabled": true
        }
    }
```

```
}
```

Cette commande renvoie l'état du transfert des commentaires par e-mail et, le cas échéant, les Amazon Resource Names (ARNs) des rubriques Amazon SNS auxquelles les notifications de renvoi, de réclamation et de livraison sont envoyées.

Si vous appelez cette commande avec une identité que vous n'avez jamais soumise pour vérification, cette identité n'apparaîtra pas dans la sortie.

Pour plus d'informations sur les notifications, consultez la section Utilisation des notifications avec Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetIdentityNotificationAttributesà la section Référence des AWS CLI commandes.

`get-identity-verification-attributes`

L'exemple de code suivant montre comment utiliser`get-identity-verification-attributes`.

AWS CLI

Pour obtenir le statut de vérification Amazon SES pour une liste d'identités

L'exemple suivant utilise la commande `get-identity-verification-attributes` pour récupérer le statut de vérification Amazon SES pour une liste d'identités :

```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

Sortie :

```
{
   "VerificationAttributes": {
       "user1@example.com": {
           "VerificationStatus": "Success"
       },
       "user2@example.com": {
           "VerificationStatus": "Pending"
       }
   }
}
```

Si vous appelez cette commande avec une identité que vous n'avez jamais soumise pour vérification, cette identité n'apparaîtra pas dans la sortie.

Pour plus d'informations sur les identités vérifiées, consultez Vérification des adresses e-mail et des domaines dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous GetIdentityVerificationAttributesà la section Référence des AWS CLI commandes.

`get-send-quota`

L'exemple de code suivant montre comment utiliser`get-send-quota`.

AWS CLI

Pour obtenir vos limites d'envoi Amazon SES

L'exemple suivant utilise la commande `get-send-quota` pour renvoyer vos limites d'envoi Amazon SES :

```
aws ses get-send-quota
```

Sortie :

```
{
   "Max24HourSend": 200.0,
   "SentLast24Hours": 1.0,
   "MaxSendRate": 1.0
}
```

Max24 HourSend est votre quota d'envoi, c'est-à-dire le nombre maximum d'e-mails que vous pouvez envoyer sur une période de 24 heures. Le quota d'envoi reflète une période glissante. Chaque fois que vous essayez d'envoyer un e-mail, Amazon SES vérifie le nombre d'e-mails que vous avez envoyés dans les dernières 24 heures. Tant que le nombre total d'e-mails que vous avez envoyés est inférieur à votre quota, votre demande d'envoi est acceptée et votre e-mail est envoyé.

SentLast24Hours est le nombre d'e-mails que vous avez envoyés au cours des 24 heures précédentes.

MaxSendRate est le nombre maximum d'e-mails que vous pouvez envoyer par seconde.

Notez que les limites d'envoi sont définies en fonction des destinataires et non pas des messages. Par exemple, un e-mail qui a 10 destinataires compte pour 10 dans votre quota d'envoi.

Pour plus d'informations, consultez Gestion de vos limites d'envoi Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous GetSendQuotaà la section Référence des AWS CLI commandes.

`get-send-statistics`

L'exemple de code suivant montre comment utiliser`get-send-statistics`.

AWS CLI

Pour obtenir les statistiques d'envoi de votre Amazon SES

L'exemple suivant utilise la `get-send-statistics` commande pour renvoyer les statistiques d'envoi de votre Amazon SES

```
aws ses get-send-statistics
```

Sortie :

```
{
   "SendDataPoints": [
       {
           "Complaints": 0,
           "Timestamp": "2013-06-12T19:32:00Z",
           "DeliveryAttempts": 2,
           "Bounces": 0,
           "Rejects": 0
       },
       {
           "Complaints": 0,
           "Timestamp": "2013-06-12T00:47:00Z",
           "DeliveryAttempts": 1,
           "Bounces": 0,
           "Rejects": 0
       }
   ]
}
```

Le résultat est une liste de points de données représentant les deux dernières semaines d'activité d'envoi. Chaque point de données de la liste contient des statistiques pour un intervalle de 15 minutes.

Dans cet exemple, il n'y a que deux points de données, car les seuls e-mails envoyés par l'utilisateur au cours des deux dernières semaines se situaient à deux intervalles de 15 minutes.

Pour plus d'informations, consultez la section Surveillance de vos statistiques d'utilisation d'Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetSendStatisticsà la section Référence des AWS CLI commandes.

`list-identities`

L'exemple de code suivant montre comment utiliser`list-identities`.

AWS CLI

Pour répertorier toutes les identités (adresses e-mail et domaines) d'un AWS compte spécifique

L'exemple suivant utilise la commande `list-identities` pour répertorier toutes les identités soumises pour vérification auprès d'Amazon SES :

```
aws ses list-identities
```

Sortie :

```
{
    "Identities": [
      "user@example.com",
      "example.com"
    ]
}
```

La liste renvoyée contient toutes les identités, quel que soit le statut de vérification (vérifié, en attente de vérification, échec, etc.).

Dans cet exemple, les adresses e-mail et les domaines sont renvoyés car nous n'avons pas spécifié le paramètre identity-type.

Pour plus d'informations sur la vérification, consultez Vérification des adresses e-mail et des domaines dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous ListIdentitiesà la section Référence des AWS CLI commandes.

`send-email`

L'exemple de code suivant montre comment utiliser`send-email`.

AWS CLI

Pour envoyer un e-mail formaté à l'aide d'Amazon SES

L'exemple suivant utilise la commande `send-email` pour envoyer un e-mail formaté :

```
aws ses send-email --from sender@example.com --destination file://destination.json
 --message file://message.json
```

Sortie :

```
{
   "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"
}
```

La destination et le message sont des structures de données JSON enregistrées dans des fichiers .json du répertoire actuel. Ces fichiers sont les suivants :

`destination.json`:

```
{
  "ToAddresses":  ["recipient1@example.com", "recipient2@example.com"],
  "CcAddresses":  ["recipient3@example.com"],
  "BccAddresses": []
}
```

`message.json`:

```
{
   "Subject": {
```

```
        "Data": "Test email sent using the AWS CLI",
        "Charset": "UTF-8"
    },
    "Body": {
        "Text": {
            "Data": "This is the message body in text format.",
            "Charset": "UTF-8"
        },
        "Html": {
            "Data": "This message body contains HTML formatting. It can, for example,
 contain links like this one: <a class=\"ulink\" href=\"http://docs.aws.amazon.com/
ses/latest/DeveloperGuide\" target=\"_blank\">Amazon SES Developer Guide</a>.",
            "Charset": "UTF-8"
        }
    }
}
```

Remplacez les adresses e-mail de l'expéditeur et du destinataire par celles que vous souhaitez utiliser. Notez que l'adresse e-mail de l'expéditeur doit être vérifiée avec Amazon SES. Jusqu'à ce que vous obteniez un accès en production pour Amazon SES, vous devez également vérifier l'adresse e-mail de chaque destinataire, sauf si le destinataire est le simulateur de boîte aux lettres Amazon SES. Pour plus d'informations sur la vérification, consultez Vérification des adresses e-mail et des domaines dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

L'ID du message dans la sortie indique que l'appel pour envoyer un e-mail a réussi.

Si vous ne recevez pas l'e-mail, vérifiez votre boîte de courrier indésirable.

Pour plus d'informations sur l'envoi d'un e-mail formaté, consultez Envoi d'e-mails formatés à l'aide de l'API Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous SendEmailà la section Référence des AWS CLI commandes.

`send-raw-email`

L'exemple de code suivant montre comment utiliser`send-raw-email`.

AWS CLI

Pour envoyer un e-mail brut à l'aide d'Amazon SES

L'exemple suivant utilise la commande `send-raw-email` pour envoyer un e-mail avec une pièce jointe TXT :

```
aws ses send-raw-email --raw-message file://message.json
```

Sortie :

```
{
   "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

Le message brut est une structure de données JSON enregistrée dans un fichier nommé `message.json` dans le répertoire actuel. Il contient les éléments suivants :

```
{
   "Data": "From: sender@example.com\nTo: recipient@example.com\nSubject: Test email
 sent using the AWS CLI (contains an attachment)\nMIME-Version: 1.0\nContent-type:
 Multipart/Mixed; boundary=\"NextPart\"\n\n--NextPart\nContent-Type: text/plain
\n\nThis is the message body.\n\n--NextPart\nContent-Type: text/plain;\nContent-
Disposition: attachment; filename=\"attachment.txt\"\n\nThis is the text in the
 attachment.\n\n--NextPart--"
}
```

Comme vous pouvez le constater, « Data » est une longue chaîne contenant l'intégralité du contenu de l'e-mail brut au format MIME, y compris une pièce jointe appelée attachment.txt.

Remplacez sender@example.com et recipient@example.com par les adresses que vous souhaitez utiliser. Notez que l'adresse e-mail de l'expéditeur doit être vérifiée avec Amazon SES. Jusqu'à ce que vous obteniez un accès en production pour Amazon SES, vous devez également vérifier l'adresse e-mail du destinataire, sauf si le destinataire est le simulateur de boîte aux lettres Amazon SES. Pour plus d'informations sur la vérification, consultez Vérification des adresses e-mail et des domaines dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

L'ID du message dans la sortie indique que l'appel send-raw-email a réussi.

Si vous ne recevez pas l'e-mail, vérifiez votre boîte de courrier indésirable.

Pour plus d'informations sur l'envoi d'un e-mail brut, consultez Envoi d'e-mails bruts à l'aide de l'API Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous SendRawEmailà la section Référence des AWS CLI commandes.

`set-identity-dkim-enabled`

L'exemple de code suivant montre comment utiliser`set-identity-dkim-enabled`.

AWS CLI

Pour activer ou désactiver Easy DKIM pour une identité vérifiée Amazon SES

L'exemple suivant utilise la `set-identity-dkim-enabled` commande pour désactiver DKIM pour une adresse e-mail vérifiée :

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Pour plus d'informations sur Easy DKIM, consultez Easy DKIM dans Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetIdentityDkimEnabledà la section Référence des AWS CLI commandes.

`set-identity-feedback-forwarding-enabled`

L'exemple de code suivant montre comment utiliser`set-identity-feedback-forwarding-enabled`.

AWS CLI

Pour activer ou désactiver le transfert des commentaires par e-mail de rebond et de réclamation pour une identité vérifiée par Amazon SES

L'exemple suivant utilise la `set-identity-feedback-forwarding-enabled` commande pour permettre à une adresse e-mail vérifiée de recevoir des notifications de rebond et de plainte par e-mail :

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --
forwarding-enabled
```

Vous devez recevoir des notifications de rebond et de réclamation via Amazon SNS ou via le transfert de commentaires par e-mail. Vous ne pouvez donc désactiver le transfert de

commentaires par e-mail que si vous sélectionnez un sujet Amazon SNS pour les notifications de rebond et de plainte.

Pour plus d'informations sur les notifications, consultez la section Utilisation des notifications avec Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetIdentityFeedbackForwardingEnabledà la section Référence des AWS CLI commandes.

`set-identity-notification-topic`

L'exemple de code suivant montre comment utiliser`set-identity-notification-topic`.

AWS CLI

Pour définir la rubrique Amazon SNS sur laquelle Amazon SES publiera les notifications de rebond, de réclamation et de and/or livraison pour une identité vérifiée

L'exemple suivant utilise la `set-identity-notification-topic` commande pour spécifier la rubrique Amazon SNS pour laquelle une adresse e-mail vérifiée recevra des notifications de rebond :

```
aws ses set-identity-notification-topic --identity user@example.com --notification-type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

Pour plus d'informations sur les notifications, consultez la section Utilisation des notifications avec Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous SetIdentityNotificationTopicà la section Référence des AWS CLI commandes.

`verify-domain-dkim`

L'exemple de code suivant montre comment utiliser`verify-domain-dkim`.

AWS CLI

Pour générer les jetons DKIM d'un domaine vérifié pour la signature DKIM avec Amazon SES

L'exemple suivant utilise la `verify-domain-dkim` commande pour générer des jetons DKIM pour un domaine qui a été vérifié auprès d'Amazon SES :

```
aws ses verify-domain-dkim --domain example.com
```

Sortie :

```
{
   "DkimTokens": [
       "EXAMPLEq76owjnks3lnluwg65scbemvw",
       "EXAMPLEi3dnsj67hstzaj673klariwx2",
       "EXAMPLEwfbtcukvimehexktmdtaz6naj"
   ]
}
```

Pour configurer DKIM, vous devez utiliser les jetons DKIM renvoyés pour mettre à jour les paramètres DNS de votre domaine avec des enregistrements CNAME pointant vers des clés publiques DKIM hébergées par Amazon SES. Pour plus d'informations, consultez Easy DKIM dans Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

- Pour plus de détails sur l'API, reportez-vous VerifyDomainDkimà la section Référence des AWS CLI commandes.

`verify-domain-identity`

L'exemple de code suivant montre comment utiliser`verify-domain-identity`.

AWS CLI

Pour vérifier un domaine avec Amazon SES

L'exemple suivant utilise la commande `verify-domain-identity` pour vérifier un domaine :

```
aws ses verify-domain-identity --domain example.com
```

Sortie :

```
{
   "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wwmvjuEXAMPLE"
}
```

Pour terminer la vérification du domaine, vous devez ajouter un enregistrement TXT avec le jeton de vérification renvoyé aux paramètres DNS de votre domaine. Pour plus d'informations,

consultez Vérification des domaines dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous VerifyDomainIdentityà la section Référence des AWS CLI commandes.

`verify-email-identity`

L'exemple de code suivant montre comment utiliser`verify-email-identity`.

AWS CLI

Pour vérifier une adresse e-mail avec Amazon SES

L'exemple suivant utilise la commande `verify-email-identity` pour vérifier une adresse e-mail :

```
aws ses verify-email-identity --email-address user@example.com
```

Avant de pouvoir envoyer un e-mail avec Amazon SES, vous devez vérifier l'adresse ou le domaine à partir desquels vous envoyez l'e-mail afin de prouver que vous en êtes le propriétaire. Si vous n'avez pas encore l'accès en production, vous devez également vérifier toutes les adresses e-mail des destinataires, à l'exception de celles fournies par le simulateur de boîte aux lettres Amazon SES.

Après avoir été verify-email-identity appelée, l'adresse e-mail recevra un e-mail de vérification. L'utilisateur doit cliquer sur le lien disponible dans l'e-mail pour terminer le processus de vérification.

Pour plus d'informations, consultez Vérification des adresses e-mail dans Amazon SES dans le Guide du développeur Amazon Simple Email Service.

- Pour plus de détails sur l'API, reportez-vous VerifyEmailIdentityà la section Référence des AWS CLI commandes.

Exemples de Shield utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Shield.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-drt-log-bucket`

L'exemple de code suivant montre comment utiliser`associate-drt-log-bucket`.

AWS CLI

Pour autoriser le DRT à accéder à un compartiment Amazon S3

L'`associate-drt-log-bucket`exemple suivant crée une association entre le DRT et le compartiment S3 spécifié. Cela permet au DRT d'accéder au bucket au nom du compte. :

```
aws shield associate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Authorize the DDo S Response Team dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateDrtLogBucketà la section Référence des AWS CLI commandes.

`associate-drt-role`

L'exemple de code suivant montre comment utiliser`associate-drt-role`.

AWS CLI

Pour autoriser le DRT à atténuer les attaques potentielles en votre nom

L'`associate-drt-role`exemple suivant crée une association entre le DRT et le rôle spécifié. Le DRT peut utiliser le rôle pour accéder au compte et le gérer.

```
aws shield associate-drt-role \
    --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Authorize the DDo S Response Team dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateDrtRoleà la section Référence des AWS CLI commandes.

`create-protection`

L'exemple de code suivant montre comment utiliser`create-protection`.

AWS CLI

Pour activer la protection AWS Shield Advanced pour une seule AWS ressource

L'`create-protection`exemple suivant active la protection Shield Advanced pour la AWS CloudFront distribution spécifiée.

```
aws shield create-protection \
    --name "Protection for CloudFront distribution" \
    --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FXOWY8
```

Sortie :

```
{
    "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Pour plus d'informations, voir Spécifiez vos ressources à protéger dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateProtectionà la section Référence des AWS CLI commandes.

`create-subscription`

L'exemple de code suivant montre comment utiliser`create-subscription`.

AWS CLI

Pour activer la protection AWS Shield Advanced pour un compte

L'`create-subscription`exemple suivant active la protection Shield Advanced pour le compte.

```
aws shield create-subscription
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting Started with AWS Shield Advanced dans le guide du développeur de AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous CreateSubscriptionà la section Référence des AWS CLI commandes.

`delete-protection`

L'exemple de code suivant montre comment utiliser`delete-protection`.

AWS CLI

Pour supprimer la protection AWS Shield Advanced d'une AWS ressource

L'`delete-protection`exemple suivant supprime la protection AWS Shield Advanced spécifiée.

```
aws shield delete-protection \
    --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Supprimer AWS Shield Advanced d'une AWS ressource dans le Guide du développeur de AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteProtectionà la section Référence des AWS CLI commandes.

`describe-attack`

L'exemple de code suivant montre comment utiliser`describe-attack`.

AWS CLI

Pour récupérer la description détaillée d'une attaque

L'`describe-attack`exemple suivant affiche les détails de l'attaque DDo S avec l'ID d'attaque spécifié. Vous pouvez obtenir une attaque IDs en exécutant la `list-attacks` commande.

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Sortie :

```
{
    "Attack": {
        "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "ResourceArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/testElb",
        "SubResources": [
            {
                "Type": "IP",
                "Id": "192.0.2.2",
                "AttackVectors": [
                    {
                        "VectorType": "SYN_FLOOD",
                        "VectorCounters": [
                            {
                                "Name": "SYN_FLOOD_BPS",
                                "Max": 982184.0,
                                "Average": 982184.0,
                                "Sum": 11786208.0,
                                "N": 12,
                                "Unit": "BPS"
                            }
                        ]
                    }
                ],
                "Counters": []
            },
            {
                "Type": "IP",
```

```
                "Id": "192.0.2.3",
                "AttackVectors": [
                    {
                        "VectorType": "SYN_FLOOD",
                        "VectorCounters": [
                            {
                                "Name": "SYN_FLOOD_BPS",
                                "Max": 982184.0,
                                "Average": 982184.0,
                                "Sum": 9821840.0,
                                "N": 10,
                                "Unit": "BPS"
                            }
                        ]
                    }
                ],
                "Counters": []
            },
            {
                "Type": "IP",
                "Id": "192.0.2.4",
                "AttackVectors": [
                    {
                        "VectorType": "SYN_FLOOD",
                        "VectorCounters": [
                            {
                                "Name": "SYN_FLOOD_BPS",
                                "Max": 982184.0,
                                "Average": 982184.0,
                                "Sum": 7857472.0,
                                "N": 8,
                                "Unit": "BPS"
                            }
                        ]
                    }
                ],
                "Counters": []
            },
            {
                "Type": "IP",
                "Id": "192.0.2.5",
                "AttackVectors": [
                    {
                        "VectorType": "SYN_FLOOD",
```

```
                    "VectorCounters": [
                        {
                            "Name": "SYN_FLOOD_BPS",
                            "Max": 982184.0,
                            "Average": 982184.0,
                            "Sum": 1964368.0,
                            "N": 2,
                            "Unit": "BPS"
                        }
                    ]
                }
            ],
            "Counters": []
        },
        {
            "Type": "IP",
            "Id": "2001:DB8::bcde:4321:8765:0:0",
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD",
                    "VectorCounters": [
                        {
                            "Name": "SYN_FLOOD_BPS",
                            "Max": 982184.0,
                            "Average": 982184.0,
                            "Sum": 1964368.0,
                            "N": 2,
                            "Unit": "BPS"
                        }
                    ]
                }
            ],
            "Counters": []
        },
        {
            "Type": "IP",
            "Id": "192.0.2.6",
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD",
                    "VectorCounters": [
                        {
                            "Name": "SYN_FLOOD_BPS",
                            "Max": 982184.0,
```

```
                            "Average": 982184.0,
                            "Sum": 1964368.0,
                            "N": 2,
                            "Unit": "BPS"
                        }
                    ]
                }
            ],
            "Counters": []
        }
    ],
    "StartTime": 1576024927.457,
    "EndTime": 1576025647.457,
    "AttackCounters": [],
    "AttackProperties": [
        {
            "AttackLayer": "NETWORK",
            "AttackPropertyIdentifier": "SOURCE_IP_ADDRESS",
            "TopContributors": [
                {
                    "Name": "198.51.100.5",
                    "Value": 2024475682
                },
                {
                    "Name": "198.51.100.8",
                    "Value": 1311380863
                },
                {
                    "Name": "203.0.113.4",
                    "Value": 900599855
                },
                {
                    "Name": "198.51.100.4",
                    "Value": 769417366
                },
                {
                    "Name": "203.1.113.13",
                    "Value": 757992847
                }
            ],
            "Unit": "BYTES",
            "Total": 92773354841
        },
        {
```

```
                "AttackLayer": "NETWORK",
                "AttackPropertyIdentifier": "SOURCE_COUNTRY",
                "TopContributors": [
                    {
                        "Name": "United States",
                        "Value": 80938161764
                    },
                    {
                        "Name": "Brazil",
                        "Value": 9929864330
                    },
                    {
                        "Name": "Netherlands",
                        "Value": 1635009446
                    },
                    {
                        "Name": "Mexico",
                        "Value": 144832971
                    },
                    {
                        "Name": "Japan",
                        "Value": 45369000
                    }
                ],
                "Unit": "BYTES",
                "Total": 92773354841
            },
            {
                "AttackLayer": "NETWORK",
                "AttackPropertyIdentifier": "SOURCE_ASN",
                "TopContributors": [
                    {
                        "Name": "12345",
                        "Value": 74953625841
                    },
                    {
                        "Name": "12346",
                        "Value": 4440087595
                    },
                    {
                        "Name": "12347",
                        "Value": 1635009446
                    },
                    {
```

```
                        "Name": "12348",
                        "Value": 1221230000
                    },
                    {
                        "Name": "12349",
                        "Value": 1199425294
                    }
                ],
                "Unit": "BYTES",
                "Total": 92755479921
            }
        ],
        "Mitigations": []
    }
}
```

Pour plus d'informations, consultez la section Révision des incidents DDo S dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeAttackà la section Référence des AWS CLI commandes.

`describe-drt-access`

L'exemple de code suivant montre comment utiliser`describe-drt-access`.

AWS CLI

Pour obtenir une description des autorisations, le DRT doit atténuer les attaques en votre nom.

L'`describe-drt-access`exemple suivant récupère le rôle et les autorisations de compartiment S3 dont dispose le DRT, qui lui permettent de répondre à des attaques potentielles en votre nom.

```
aws shield describe-drt-access
```

Sortie :

```
{
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
    "LogBucketList": [
        "flow-logs-for-website-lb"
```

```
    ]
}
```

Pour plus d'informations, consultez Authorize the DDo S Response Team dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeDrtAccessà la section Référence des AWS CLI commandes.

`describe-emergency-contact-settings`

L'exemple de code suivant montre comment utiliser`describe-emergency-contact-settings`.

AWS CLI

Pour récupérer les adresses e-mail d'urgence que vous avez enregistrées auprès de la DRT

L'`describe-emergency-contact-settings`exemple suivant récupère les adresses e-mail enregistrées auprès du DRT pour le compte. Voici les adresses que le DRT doit contacter lorsqu'il répond à une attaque présumée.

```
aws shield describe-emergency-contact-settings
```

Sortie :

```
{
    "EmergencyContactList": [
        {
            "EmailAddress": "ops@example.com"
        },
        {
            "EmailAddress": "ddos-notifications@example.com"
       }
    ]
}
```

Pour plus d'informations, consultez How AWS Shield Works< https://docs.aws.amazon.com/waf/latest/developerguide/ddos -overview.html> dans le Shield Advanced Developer Guide.AWS

- Pour plus de détails sur l'API, reportez-vous DescribeEmergencyContactSettingsà la section Référence des AWS CLI commandes.

`describe-protection`

L'exemple de code suivant montre comment utiliser`describe-protection`.

AWS CLI

Pour récupérer les informations relatives à une protection AWS Shield Advanced

L'`describe-protection`exemple suivant affiche des détails sur la protection Shield Advanced avec l'ID spécifié. Vous pouvez obtenir une protection IDs en exécutant la `list-protections` commande.

```
aws shield describe-protection \
    --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "Protection": {
        "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "1.2.3.4",
        "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/
eipalloc-0ac1537af40742a6d"
    }
}
```

Pour plus d'informations, voir Spécifiez vos ressources à protéger dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeProtectionà la section Référence des AWS CLI commandes.

`describe-subscription`

L'exemple de code suivant montre comment utiliser`describe-subscription`.

AWS CLI

Pour récupérer les détails de la protection AWS Shield Advanced pour le compte

L'`describe-subscription`exemple suivant affiche des détails sur la protection Shield Advanced fournie pour le compte. :

```
aws shield describe-subscription
```

Sortie :

```
{
    "Subscription": {
        "StartTime": 1534368978.0,
        "EndTime": 1597613778.0,
        "TimeCommitmentInSeconds": 63244800,
        "AutoRenew": "ENABLED",
        "Limits": [
            {
                "Type": "GLOBAL_ACCELERATOR",
                "Max": 1000
            },
            {
                "Type": "ROUTE53_HOSTED_ZONE",
                "Max": 1000
            },
            {
                "Type": "CF_DISTRIBUTION",
                "Max": 1000
            },
            {
                "Type": "ELB_LOAD_BALANCER",
                "Max": 1000
            },
            {
                "Type": "EC2_ELASTIC_IP_ALLOCATION",
                "Max": 1000
            }
        ]
    }
}
```

Pour plus d'informations, consultez How AWS Shield Works dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeSubscriptionà la section Référence des AWS CLI commandes.

`disassociate-drt-log-bucket`

L'exemple de code suivant montre comment utiliser`disassociate-drt-log-bucket`.

AWS CLI

Pour supprimer l'autorisation permettant à DRT d'accéder à un compartiment Amazon S3 en votre nom

L'`disassociate-drt-log-bucket`exemple suivant supprime l'association entre le DRT et le compartiment S3 spécifié. Une fois cette commande terminée, le DRT ne peut plus accéder au bucket au nom du compte.

```
aws shield disassociate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Authorize the DDo S Response Team dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DisassociateDrtLogBucketà la section Référence des AWS CLI commandes.

`disassociate-drt-role`

L'exemple de code suivant montre comment utiliser`disassociate-drt-role`.

AWS CLI

Pour supprimer l'autorisation accordée à DRT afin d'atténuer les attaques potentielles en votre nom

L'`disassociate-drt-role`exemple suivant supprime l'association entre le DRT et le compte. Après cet appel, le DRT ne pourra plus accéder à votre compte ni le gérer.

```
aws shield disassociate-drt-role
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Authorize the DDo S Response Team dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DisassociateDrtRoleà la section Référence des AWS CLI commandes.

`get-subscription-state`

L'exemple de code suivant montre comment utiliser`get-subscription-state`.

AWS CLI

Pour connaître l'état actuel de l'abonnement AWS Shield Advanced du compte

L'`get-subscription-state`exemple suivant permet de récupérer l'état de la protection Shield Advanced pour le compte.

```
aws shield get-subscription-state
```

Sortie :

```
{
    "SubscriptionState": "ACTIVE"
}
```

Pour plus d'informations, consultez How AWS Shield Works dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetSubscriptionStateà la section Référence des AWS CLI commandes.

`list-attacks`

L'exemple de code suivant montre comment utiliser`list-attacks`.

AWS CLI

Pour récupérer les résumés des attaques depuis AWS Shield Advanced

L'`list-attacks`exemple suivant récupère les résumés des attaques pour la AWS CloudFront distribution spécifiée au cours de la période spécifiée. La réponse inclut une attaque IDs que vous pouvez fournir à la `describe-attack` commande pour obtenir des informations détaillées sur une attaque.

```
aws shield list-attacks \
    --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXMP22ZVFAOR \
    --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

Sortie :

```
{
    "AttackSummaries": [
        {
            "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E1PXMP22ZVFAOR",
            "StartTime": 1529280000.0,
            "EndTime": 1529449200.0,
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Révision des incidents DDo S dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListAttacksà la section Référence des AWS CLI commandes.

`list-protections`

L'exemple de code suivant montre comment utiliser`list-protections`.

AWS CLI

Pour récupérer les résumés de protection depuis AWS Shield Advanced

L'`list-protections`exemple suivant permet de récupérer des résumés des protections activées pour le compte.

```
aws shield list-protections
```

Sortie :

```
{
    "Protections": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Name": "Protection for CloudFront distribution",
            "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FXOWY8"
        }
    ]
}
```

Pour plus d'informations, voir Spécifiez vos ressources à protéger dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListProtectionsà la section Référence des AWS CLI commandes.

`update-emergency-contact-settings`

L'exemple de code suivant montre comment utiliser`update-emergency-contact-settings`.

AWS CLI

Pour définir les adresses e-mail d'urgence enregistrées auprès du DRT

L'`update-emergency-contact-settings`exemple suivant définit deux adresses e-mail que le DRT doit contacter lorsqu'il répond à une attaque présumée.

```
aws shield update-emergency-contact-settings \
        --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-
notifications@example.com
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez How AWS Shield Works dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateEmergencyContactSettingsà la section Référence des AWS CLI commandes.

`update-subscription`

L'exemple de code suivant montre comment utiliser`update-subscription`.

AWS CLI

Pour modifier l'abonnement AWS Shield Advanced du compte

L'`update-subscription`exemple suivant active le renouvellement automatique de l'abonnement AWS Shield Advanced pour le compte.

```
aws shield update-subscription \
    --auto-renew ENABLED
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez How AWS Shield Works dans le AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateSubscriptionà la section Référence des AWS CLI commandes.

Exemples de signataires utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Signer.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`cancel-signing-profile`

L'exemple de code suivant montre comment utiliser`cancel-signing-profile`.

AWS CLI

Pour supprimer un profil de signature

L'`cancel-signing-profile`exemple suivant supprime un profil de signature existant de AWS Signer.

```
aws signer cancel-signing-profile \
    --profile-name MyProfile1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous CancelSigningProfileà la section Référence des AWS CLI commandes.

`describe-signing-job`

L'exemple de code suivant montre comment utiliser`describe-signing-job`.

AWS CLI

Pour afficher les détails d'une tâche de signature

L'`describe-signing-job`exemple suivant affiche les détails de la tâche de signature spécifiée.

```
aws signer describe-signing-job \
    --job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

Sortie :

```
{
    "status": "Succeeded",
    "completedAt": 1568412037,
    "platformId": "AmazonFreeRTOS-Default",
    "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
```

```
    },
    "statusReason": "Signing Succeeded",
    "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
    "source": {
        "s3": {
            "version": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4",
            "bucketName": "signer-source",
            "key": "MyCode.rb"
        }
    },
    "profileName": "MyProfile2",
    "signedObject": {
        "s3": {
            "bucketName": "signer-destination",
            "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
        }
    },
    "requestedBy": "arn:aws:iam::123456789012:user/maria",
    "createdAt": 1568412036
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSigningJobà la section Référence des AWS CLI commandes.

`get-signing-platform`

L'exemple de code suivant montre comment utiliser`get-signing-platform`.

AWS CLI

Pour afficher les informations relatives à une plateforme de signature

L'`get-signing-platform`exemple suivant affiche des détails sur la plate-forme de signature spécifiée.

```
aws signer get-signing-platform \
    --platform-id AmazonFreeRTOS-TI-CC3220SF
```

Sortie :

```
{
```

```
    "category": "AWS",
    "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
    "target": "SHA1-RSA-TISHA1",
    "platformId": "AmazonFreeRTOS-TI-CC3220SF",
    "signingConfiguration": {
        "encryptionAlgorithmOptions": {
            "defaultValue": "RSA",
            "allowedValues": [
                "RSA"
            ]
        },
        "hashAlgorithmOptions": {
            "defaultValue": "SHA1",
            "allowedValues": [
                "SHA1"
            ]
        }
    },
    "maxSizeInMB": 16,
    "partner": "AmazonFreeRTOS",
    "signingImageFormat": {
        "defaultFormat": "JSONEmbedded",
        "supportedFormats": [
            "JSONEmbedded"
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSigningPlatformà la section Référence des AWS CLI commandes.

`get-signing-profile`

L'exemple de code suivant montre comment utiliser`get-signing-profile`.

AWS CLI

Pour afficher les informations relatives à un profil de signature

L'`get-signing-profile`exemple suivant affiche les détails du profil de signature spécifié.

```
aws signer get-signing-profile \
```

```
    --profile-name MyProfile3
```

Sortie :

```
{
    "platformId": "AmazonFreeRTOS-TI-CC3220SF",
    "profileName": "MyProfile3",
    "status": "Active",
    "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSigningProfileà la section Référence des AWS CLI commandes.

`list-signing-jobs`

L'exemple de code suivant montre comment utiliser`list-signing-jobs`.

AWS CLI

Pour répertorier toutes les tâches de signature

L'`list-signing-jobs`exemple suivant affiche les détails de toutes les tâches de signature pour le compte.

```
aws signer list-signing-jobs
```

Dans cet exemple, deux tâches sont renvoyées, l'une réussie et l'autre ayant échoué.

```
{
    "jobs": [
        {
            "status": "Succeeded",
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            },
            "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
            "source": {
```

```
                "s3": {
                    "version": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4",
                    "bucketName": "signer-source",
                    "key": "MyCode.rb"
                }
            },
            "signedObject": {
                "s3": {
                    "bucketName": "signer-destination",
                    "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
                }
            },
            "createdAt": 1568412036
        },
        {
            "status": "Failed",
            "source": {
                "s3": {
                    "version": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4",
                    "bucketName": "signer-source",
                    "key": "MyOtherCode.rb"
                }
            },
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            },
            "createdAt": 1568402690,
            "jobId": "74d9825e-22fc-4a0d-b962-0123456789abc"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSigningJobsà la section Référence des AWS CLI commandes.

`list-signing-platforms`

L'exemple de code suivant montre comment utiliser`list-signing-platforms`.

AWS CLI

Pour répertorier toutes les plateformes de signature

L'`list-signing-platforms`exemple suivant affiche des informations détaillées sur toutes les plateformes de signature disponibles.

```
aws signer list-signing-platforms
```

Sortie :

```
{
    "platforms": [
        {
            "category": "AWS",
            "displayName": "AWS IoT Device Management SHA256-ECDSA ",
            "target": "SHA256-ECDSA",
            "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
            "signingConfiguration": {
                "encryptionAlgorithmOptions": {
                    "defaultValue": "ECDSA",
                    "allowedValues": [
                        "ECDSA"
                    ]
                },
                "hashAlgorithmOptions": {
                    "defaultValue": "SHA256",
                    "allowedValues": [
                        "SHA256"
                    ]
                }
            },
            "maxSizeInMB": 2048,
            "partner": "AWSIoTDeviceManagement",
            "signingImageFormat": {
                "defaultFormat": "JSONDetached",
                "supportedFormats": [
                    "JSONDetached"
                ]
            }
        },
        {
            "category": "AWS",
            "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
            "target": "SHA1-RSA-TISHA1",
            "platformId": "AmazonFreeRTOS-TI-CC3220SF",
            "signingConfiguration": {
```

```
                "encryptionAlgorithmOptions": {
                    "defaultValue": "RSA",
                    "allowedValues": [
                        "RSA"
                    ]
                },
                "hashAlgorithmOptions": {
                    "defaultValue": "SHA1",
                    "allowedValues": [
                        "SHA1"
                    ]
                }
            },
            "maxSizeInMB": 16,
            "partner": "AmazonFreeRTOS",
            "signingImageFormat": {
                "defaultFormat": "JSONEmbedded",
                "supportedFormats": [
                    "JSONEmbedded"
                ]
            }
        },
        {
            "category": "AWS",
            "displayName": "Amazon FreeRTOS SHA256-ECDSA",
            "target": "SHA256-ECDSA",
            "platformId": "AmazonFreeRTOS-Default",
            "signingConfiguration": {
                "encryptionAlgorithmOptions": {
                    "defaultValue": "ECDSA",
                    "allowedValues": [
                        "ECDSA"
                    ]
                },
                "hashAlgorithmOptions": {
                    "defaultValue": "SHA256",
                    "allowedValues": [
                        "SHA256"
                    ]
                }
            },
            "maxSizeInMB": 16,
            "partner": "AmazonFreeRTOS",
            "signingImageFormat": {
```

```
                "defaultFormat": "JSONEmbedded",
                "supportedFormats": [
                    "JSONEmbedded"
                ]
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSigningPlatformsà la section Référence des AWS CLI commandes.

`list-signing-profiles`

L'exemple de code suivant montre comment utiliser`list-signing-profiles`.

AWS CLI

Pour répertorier tous les profils de signature

L'`list-signing-profiles`exemple suivant affiche les détails de tous les profils de signature du compte.

```
aws signer list-signing-profiles
```

Sortie :

```
{
    "profiles": [
        {
            "platformId": "AmazonFreeRTOS-TI-CC3220SF",
            "profileName": "MyProfile4",
            "status": "Active",
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            }
        },
        {
            "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
            "profileName": "MyProfile5",
            "status": "Active",
```

```
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSigningProfilesà la section Référence des AWS CLI commandes.

`put-signing-profile`

L'exemple de code suivant montre comment utiliser`put-signing-profile`.

AWS CLI

Pour créer un profil de signature

L'`put-signing-profile`exemple suivant crée un profil de signature à l'aide du certificat et de la plate-forme spécifiés.

```
aws signer put-signing-profile \
    --profile-name MyProfile6 \
    --signing-material certificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc \
    --platform AmazonFreeRTOS-TI-CC3220SF
```

Sortie :

```
{
    "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"
}
```

- Pour plus de détails sur l'API, reportez-vous PutSigningProfileà la section Référence des AWS CLI commandes.

`start-signing-job`

L'exemple de code suivant montre comment utiliser`start-signing-job`.

AWS CLI

Pour démarrer une tâche de signature

L'`start-signing-job`exemple suivant lance une tâche de signature sur le code trouvé dans la source spécifiée. Il utilise le profil spécifié pour effectuer la signature et place le code signé dans la destination spécifiée.

```
aws signer start-signing-job \
    --source 's3={bucketName=signer-
source,key=MyCode.rb,version=PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4}' \
    --destination 's3={bucketName=signer-destination,prefix=signed-}' \
    --profile-name MyProfile7
```

Le résultat est l'ID de la tâche de signature.

```
{
    "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"
}
```

- Pour plus de détails sur l'API, reportez-vous StartSigningJobà la section Référence des AWS CLI commandes.

Exemples d'utilisation de Snowball Edge AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide de Snowball Edge.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-snowball-usage`

L'exemple de code suivant montre comment utiliser`get-snowball-usage`.

AWS CLI

Pour obtenir des informations sur la limite de service Snowball pour votre compte

L'`get-snowball-usage`exemple suivant affiche des informations sur la limite de service Snowball pour votre compte, ainsi que sur le nombre de Snowballs que votre compte utilise.

```
aws snowball get-snowball-usage
```

Sortie :

```
{
    "SnowballLimit": 1,
    "SnowballsInUse": 0
}
```

POUR plus d'informations, consultez les limites de AWS Snowball Edge dans le guide du développeur de AWS Snowball.

- Pour plus de détails sur l'API, reportez-vous GetSnowballUsageà la section Référence des AWS CLI commandes.

`list-jobs`

L'exemple de code suivant montre comment utiliser`list-jobs`.

AWS CLI

Pour répertorier les offres d'emploi actuelles de Snowball sur votre compte

L'`list-jobs`exemple suivant affiche un tableau d'`JobListEntry`objets. Dans cet exemple, une seule tâche est répertoriée.

```
aws snowball list-jobs
```

Sortie :

```
{
    "JobListEntries": [
        {
            "CreationDate": 2016-09-27T14:50Z,
             "Description": "Important Photos 2016-08-11",
             "IsMaster": TRUE,
             "JobId": "ABCd1e324fe-022f-488e-a98b-3b0566063db1",
             "JobState": "Complete",
             "JobType": "IMPORT",
             "SnowballType": "EDGE"
        }
   ]
}
```

Pour plus d'informations, consultez la section Jobs for AWS Snowball Edge dans le manuel AWS Snowball Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListJobsà la section Référence des AWS CLI commandes.

Exemples d'utilisation d'Amazon SNS AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon SNS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions
- Scénarios

Actions

`add-permission`

L'exemple de code suivant montre comment utiliser`add-permission`.

AWS CLI

Pour ajouter une autorisation à un sujet

L'`add-permission`exemple suivant ajoute l'autorisation `987654321098` au AWS compte d'utiliser l'`Publish`action avec le sujet spécifié sous AWS compte`123456789012`.

```
aws sns add-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission \
    --aws-account-id 987654321098 \
    --action-name Publish
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AddPermissionà la section Référence des AWS CLI commandes.

`check-if-phone-number-is-opted-out`

L'exemple de code suivant montre comment utiliser`check-if-phone-number-is-opted-out`.

AWS CLI

Pour vérifier la désactivation d'un numéro de téléphone aux SMS

L'`check-if-phone-number-is-opted-out`exemple suivant vérifie si le numéro de téléphone spécifié est désactivé pour ne pas recevoir de SMS en provenance du AWS compte courant.

```
aws sns check-if-phone-number-is-opted-out \
    --phone-number +1555550100
```

Sortie :

```
{
    "isOptedOut": false
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous CheckIfPhoneNumberIsOptedOutà la section Référence des AWS CLI commandes.

`confirm-subscription`

L'exemple de code suivant montre comment utiliser`confirm-subscription`.

AWS CLI

Pour confirmer un abonnement

La commande `confirm-subscription` suivante termine le processus de confirmation lancé lorsque vous vous êtes abonné à une rubrique SNS nommée `my-topic`. Le paramètre --token provient du message de confirmation envoyé au point de terminaison de notification spécifié dans l'appel d'abonnement.

```
aws sns confirm-subscription \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
    --
token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5d
```

Sortie :

```
{
    "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
}
```

- Pour plus de détails sur l'API, reportez-vous ConfirmSubscriptionà la section Référence des AWS CLI commandes.

`create-platform-application`

L'exemple de code suivant montre comment utiliser`create-platform-application`.

AWS CLI

Pour créer une application de plateforme

L'`create-platform-application`exemple suivant crée une application de plate-forme Google Firebase à l'aide des informations d'identification de plate-forme spécifiées.

```
aws sns create-platform-application \
    --name MyApplication \
    --platform GCM \
    --attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

Sortie :

```
{
    "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication"
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePlatformApplicationà la section Référence des AWS CLI commandes.

`create-topic`

L'exemple de code suivant montre comment utiliser`create-topic`.

AWS CLI

Pour créer une rubrique SNS

L'exemple `create-topic` suivant crée une rubrique SNS nommée `my-topic`.

```
aws sns create-topic \
    --name my-topic
```

Sortie :

```
{
    "ResponseMetadata": {
        "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"
    },
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

Pour plus d'informations, consultez la section Utilisation de l'interface de ligne de AWS commande avec Amazon SQS et Amazon SNS dans le Guide de l'utilisateur de AWS l'interface de ligne de commande.

- Pour plus de détails sur l'API, reportez-vous CreateTopicà la section Référence des AWS CLI commandes.

`delete-endpoint`

L'exemple de code suivant montre comment utiliser`delete-endpoint`.

AWS CLI

Pour supprimer un point de terminaison d'une application de plateforme

L'`delete-endpoint`exemple suivant supprime le point de terminaison de l'application de plate-forme spécifié.

```
aws sns delete-endpoint \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteEndpointà la section Référence des AWS CLI commandes.

`delete-platform-application`

L'exemple de code suivant montre comment utiliser`delete-platform-application`.

AWS CLI

Pour supprimer une application de plateforme

L'`delete-platform-application`exemple suivant supprime l'application de plate-forme spécifiée.

```
aws sns delete-platform-application \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/
MyApplication
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePlatformApplicationà la section Référence des AWS CLI commandes.

`delete-topic`

L'exemple de code suivant montre comment utiliser`delete-topic`.

AWS CLI

Pour supprimer une rubrique SNS

L'exemple `delete-topic` suivant supprime la rubrique SNS spécifiée.

```
aws sns delete-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteTopicà la section Référence des AWS CLI commandes.

`get-endpoint-attributes`

L'exemple de code suivant montre comment utiliser`get-endpoint-attributes`.

AWS CLI

Pour répertorier les attributs des points de terminaison d'une application

L'`get-endpoint-attributes`exemple suivant répertorie les attributs du point de terminaison de l'application de plate-forme spécifié.

```
aws sns get-endpoint-attributes \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234
```

Sortie :

```
{
```

```
    "Attributes": {
        "Enabled": "true",
        "Token": "EXAMPLE12345..."
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetEndpointAttributesà la section Référence des AWS CLI commandes.

`get-platform-application-attributes`

L'exemple de code suivant montre comment utiliser`get-platform-application-attributes`.

AWS CLI

Pour répertorier les attributs de l'application de plate-forme

L'`get-platform-application-attributes`exemple suivant répertorie les attributs de l'application de plate-forme spécifiée.

```
aws sns get-platform-application-attributes \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/
MyApplication
```

Sortie :

```
{
    "Attributes": {
        "Enabled": "true",
        "SuccessFeedbackSampleRate": "100"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPlatformApplicationAttributesà la section Référence des AWS CLI commandes.

`get-sms-attributes`

L'exemple de code suivant montre comment utiliser`get-sms-attributes`.

AWS CLI

Pour répertorier les attributs des SMS par défaut

L'exemple `get-sms-attributes` suivant répertorie les attributs par défaut pour l'envoi de SMS.

```
aws sns get-sms-attributes
```

Sortie :

```
{
    "attributes": {
        "DefaultSenderID": "MyName"
    }
}
```

- Pour plus de détails sur l'API, voir Get SMSAttributes in AWS CLI Command Reference.

`get-subscription-attributes`

L'exemple de code suivant montre comment utiliser`get-subscription-attributes`.

AWS CLI

Pour récupérer les attributs d'abonnement pour une rubrique

Les attributs de l'abonnement spécifié sont `get-subscription-attributes` affichés ci-dessous. Vous pouvez l'obtenir `subscription-arn` à partir de la sortie de la `list-subscriptions` commande.

```
aws sns get-subscription-attributes \
    --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

Sortie :

```
{
    "Attributes": {
        "Endpoint": "my-email@example.com",
        "Protocol": "email",
        "RawMessageDelivery": "false",
        "ConfirmationWasAuthenticated": "false",
```

```
        "Owner": "123456789012",
        "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f",
        "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSubscriptionAttributesà la section Référence des AWS CLI commandes.

`get-topic-attributes`

L'exemple de code suivant montre comment utiliser`get-topic-attributes`.

AWS CLI

Pour récupérer les attributs d'une rubrique

L'exemple `get-topic-attributes` suivant affiche les attributs de la rubrique spécifiée.

```
aws sns get-topic-attributes \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Sortie :

```
{
    "Attributes": {
        "SubscriptionsConfirmed": "1",
        "DisplayName": "my-topic",
        "SubscriptionsDeleted": "0",
        "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":
{\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries
\":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear
\"},\"disableSubscriptionOverrides\":false}}",
        "Owner": "123456789012",
        "Policy": "{\"Version\":\"2008-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic
\",\"SNS:DeleteTopic\",\"SNS:GetTopicAttributes\",\"SNS:Publish\",
\"SNS:RemovePermission\",\"SNS:AddPermission\",\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-west-2:123456789012:my-topic\",\"Condition\":
{\"StringEquals\":{\"AWS:SourceOwner\":\"0123456789012\"}}}]}",
```

```
        "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
        "SubscriptionsPending": "0"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetTopicAttributesà la section Référence des AWS CLI commandes.

`list-endpoints-by-platform-application`

L'exemple de code suivant montre comment utiliser`list-endpoints-by-platform-application`.

AWS CLI

Pour répertorier les points de terminaison d'une application de plateforme

L'`list-endpoints-by-platform-application`exemple suivant répertorie les points de terminaison et les attributs de point de terminaison pour l'application de plate-forme spécifiée.

```
aws sns list-endpoints-by-platform-application \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication
```

Sortie :

```
{
    "Endpoints": [
        {
            "Attributes": {
                "Token": "EXAMPLE12345...,
                "Enabled": "true"
            },
            "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListEndpointsByPlatformApplicationà la section Référence des AWS CLI commandes.

`list-phone-numbers-opted-out`

L'exemple de code suivant montre comment utiliser`list-phone-numbers-opted-out`.

AWS CLI

Pour répertorier les désactivations des SMS

L'exemple `list-phone-numbers-opted-out` suivant répertorie les numéros de téléphone qui ont désactivé la réception de SMS.

```
aws sns list-phone-numbers-opted-out
```

Sortie :

```
{
    "phoneNumbers": [
        "+15555550100"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPhoneNumbersOptedOutà la section Référence des AWS CLI commandes.

`list-platform-applications`

L'exemple de code suivant montre comment utiliser`list-platform-applications`.

AWS CLI

Pour répertorier les applications de la plateforme

L'`list-platform-applications`exemple suivant répertorie les applications de plate-forme pour ADM et MPNS.

```
aws sns list-platform-applications
```

Sortie :

```
{
    "PlatformApplications": [
```

```
        {
            "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/ADM/
MyApplication",
            "Attributes": {
                "SuccessFeedbackSampleRate": "100",
                "Enabled": "true"
            }
        },
        {
            "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/MPNS/
MyOtherApplication",
            "Attributes": {
                "SuccessFeedbackSampleRate": "100",
                "Enabled": "true"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListPlatformApplicationsà la section Référence des AWS CLI commandes.

`list-subscriptions-by-topic`

L'exemple de code suivant montre comment utiliser`list-subscriptions-by-topic`.

AWS CLI

Pour répertorier les abonnements associés à un sujet

Ce qui suit `list-subscriptions-by-topic` permet de récupérer la liste des abonnements SNS associés à la rubrique spécifiée.

```
aws sns list-subscriptions-by-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Sortie :

```
{
    "Subscriptions": [
        {
            "Owner": "123456789012",
```

```
            "Endpoint": "my-email@example.com",
            "Protocol": "email",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
            "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSubscriptionsByTopicà la section Référence des AWS CLI commandes.

`list-subscriptions`

L'exemple de code suivant montre comment utiliser`list-subscriptions`.

AWS CLI

Pour répertorier vos abonnements SNS

L'`list-subscriptions`exemple suivant affiche la liste des abonnements SNS de votre AWS compte.

```
aws sns list-subscriptions
```

Sortie :

```
{
    "Subscriptions": [
        {
            "Owner": "123456789012",
            "Endpoint": "my-email@example.com",
            "Protocol": "email",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
            "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListSubscriptionsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les tags d'un sujet

L'`list-tags-for-resource`exemple suivant répertorie les balises pour la rubrique Amazon SNS spécifiée.

```
aws sns list-tags-for-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "Team",
            "Value": "Alpha"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-topics`

L'exemple de code suivant montre comment utiliser`list-topics`.

AWS CLI

Pour répertorier vos rubriques SNS

L'`list-topics`exemple suivant répertorie toutes les rubriques SNS de votre AWS compte.

```
aws sns list-topics
```

Sortie :

```
{
    "Topics": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListTopicsà la section Référence des AWS CLI commandes.

`opt-in-phone-number`

L'exemple de code suivant montre comment utiliser`opt-in-phone-number`.

AWS CLI

Pour vous inscrire aux messages SMS

L'`opt-in-phone-number`exemple suivant permet de choisir le numéro de téléphone spécifié pour recevoir des SMS.

```
aws sns opt-in-phone-number \
    --phone-number +15555550100
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous OptInPhoneNumberà la section Référence des AWS CLI commandes.

`publish`

L'exemple de code suivant montre comment utiliser`publish`.

AWS CLI

Exemple 1 : pour publier un message dans une rubrique

L'exemple `publish` suivant publie le message spécifié sur la rubrique SNS spécifiée. Le message provient d'un fichier texte qui vous permet d'inclure des sauts de ligne.

```
aws sns publish \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \
    --message file://message.txt
```

Contenu de `message.txt` :

```
Hello World
Second Line
```

Sortie :

```
{
    "MessageId": "123a45b6-7890-12c3-45d6-111122223333"
}
```

Exemple 2 : pour publier un SMS sur un numéro de téléphone

L'exemple `publish` suivant publie le message `Hello world!` sur le numéro de téléphone `+1-555-555-0100`.

```
aws sns publish \
    --message "Hello world!" \
    --phone-number +1-555-555-0100
```

Sortie :

```
{
    "MessageId": "123a45b6-7890-12c3-45d6-333322221111"
}
```

- Pour plus d'informations sur l'API, consultez Publish dans la Référence des commandes AWS CLI .

`put-data-protection-policy`

L'exemple de code suivant montre comment utiliser`put-data-protection-policy`.

AWS CLI

Pour définir la politique de protection des données

Exemple 1 : pour empêcher les éditeurs de publier des messages avec CreditCardNumber

L'`put-data-protection-policy`exemple suivant interdit aux éditeurs de publier des messages avec CreditCardNumber.

```
aws sns put-data-protection-policy \
    --resource-arn arn:aws:sns:us-east-1:123456789012:mytopic \
    --data-protection-policy "{\"Name\":\"data_protection_policy\",\"Description
\":\"Example data protection policy\",\"Version\":\"2021-06-01\",\"Statement
\":[{\"DataDirection\":\"Inbound\",\"Principal\":[\"*\"],\"DataIdentifier\":
[\"arn:aws:dataprotection::aws:data-identifier/CreditCardNumber\"],\"Operation\":
{\"Deny\":{}}}]}"
```

Cette commande ne produit aucun résultat.

Exemple 2 : pour charger des paramètres depuis un fichier

Ce qui suit `put-data-protection-policy` charge les paramètres à partir d'un fichier.

```
aws sns put-data-protection-policy \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --data-protection-policy file://policy.json
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutDataProtectionPolicyà la section Référence des AWS CLI commandes.

`remove-permission`

L'exemple de code suivant montre comment utiliser`remove-permission`.

AWS CLI

Pour supprimer une autorisation d'un sujet

L'`remove-permission`exemple suivant supprime l'autorisation `Publish-Permission` du sujet spécifié.

```
aws sns remove-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RemovePermissionà la section Référence des AWS CLI commandes.

`set-endpoint-attributes`

L'exemple de code suivant montre comment utiliser`set-endpoint-attributes`.

AWS CLI

Pour définir les attributs du point de terminaison

L'`set-endpoint-attributes`exemple suivant désactive le point de terminaison de l'application de plate-forme spécifié.

```
aws sns set-endpoint-attributes \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234 \
    --attributes Enabled=false
```

Sortie :

```
{
    "Attributes": {
        "Enabled": "false",
        "Token": "EXAMPLE12345..."
    }
}
```

- Pour plus de détails sur l'API, reportez-vous SetEndpointAttributesà la section Référence des AWS CLI commandes.

`set-platform-application-attributes`

L'exemple de code suivant montre comment utiliser`set-platform-application-attributes`.

AWS CLI

Pour définir les attributs de l'application de plate-forme

L'`set-platform-application-attributes`exemple suivant définit l'`EventDeliveryFailure`attribut de l'application de plate-forme spécifiée sur l'ARN de la rubrique Amazon SNS spécifiée.

```
aws sns set-platform-application-attributes \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication \
    --attributes EventDeliveryFailure=arn:aws:sns:us-
west-2:123456789012:AnotherTopic
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetPlatformApplicationAttributesà la section Référence des AWS CLI commandes.

`set-sms-attributes`

L'exemple de code suivant montre comment utiliser`set-sms-attributes`.

AWS CLI

Pour définir les attributs des SMS

L'exemple `set-sms-attributes` suivant définit l'ID d'expéditeur par défaut des SMS sur `MyName`.

```
aws sns set-sms-attributes \
    --attributes DefaultSenderID=MyName
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, voir Set SMSAttributes in AWS CLI Command Reference.

`set-subscription-attributes`

L'exemple de code suivant montre comment utiliser`set-subscription-attributes`.

AWS CLI

Pour définir des attributs d'abonnement

L'exemple `set-subscription-attributes` suivant définit l'attribut `RawMessageDelivery` sur un abonnement SQS.

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name RawMessageDelivery \
    --attribute-value true
```

Cette commande ne produit aucun résultat.

L'exemple `set-subscription-attributes` suivant définit un attribut `FilterPolicy` sur un abonnement SQS.

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name FilterPolicy \
    --attribute-value "{ \"anyMandatoryKey\": [\"any\", \"of\", \"these\"] }"
```

Cette commande ne produit aucun résultat.

L'exemple `set-subscription-attributes` suivant supprime l'attribut `FilterPolicy` d'un abonnement SQS.

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name FilterPolicy \
    --attribute-value "{}"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetSubscriptionAttributesà la section Référence des AWS CLI commandes.

`set-topic-attributes`

L'exemple de code suivant montre comment utiliser`set-topic-attributes`.

AWS CLI

Pour définir un attribut pour une rubrique

L'exemple `set-topic-attributes` suivant définit l'attribut `DisplayName` pour la rubrique spécifiée.

```
aws sns set-topic-attributes \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --attribute-name DisplayName \
    --attribute-value MyTopicDisplayName
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SetTopicAttributesà la section Référence des AWS CLI commandes.

`subscribe`

L'exemple de code suivant montre comment utiliser`subscribe`.

AWS CLI

Pour s'abonner à une rubrique

La commande `subscribe` suivante abonne une adresse e-mail à la rubrique spécifiée.

```
aws sns subscribe \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
    --protocol email \
    --notification-endpoint my-email@example.com
```

Sortie :

```
{
    "SubscriptionArn": "pending confirmation"
}
```

- Pour plus d'informations sur l'API, consultez Subscribe dans la Référence des commandes AWS CLI .

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter une balise à une rubrique

L'exemple `tag-resource` suivant ajoute une balise de métadonnées à la rubrique Amazon SNS spécifiée.

```
aws sns tag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tags Key=Team,Value=Alpha
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`unsubscribe`

L'exemple de code suivant montre comment utiliser`unsubscribe`.

AWS CLI

Pour se désabonner d'une rubrique

L'exemple `unsubscribe` suivant supprime l'abonnement spécifié d'une rubrique.

```
aws sns unsubscribe \
    --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

Cette commande ne produit aucun résultat.

- Pour plus d'informations sur l'API, consultez Unsubscribe dans la Référence des commandes AWS CLI .

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer un tag d'un sujet

L'`untag-resource`exemple suivant supprime toutes les balises contenant les clés spécifiées de la rubrique Amazon SNS spécifiée.

```
aws sns untag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tag-keys Team
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

Scénarios

Créer un point de terminaison de plateforme pour les notifications push

L'exemple de code suivant montre comment créer un point de terminaison de plateforme pour les notifications push Amazon SNS.

AWS CLI

Pour créer un point de terminaison d'application de plateforme

L'exemple `create-platform-endpoint` suivant crée un point de terminaison pour l'application de plateforme spécifiée à l'aide du jeton spécifié.

```
aws sns create-platform-endpoint \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication \
    --token EXAMPLE12345...
```

Sortie :

```
{
      "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
}
```

Exemples d'utilisation d'Amazon SQS AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon SQS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-permission`

L'exemple de code suivant montre comment utiliser`add-permission`.

AWS CLI

Pour ajouter une autorisation à une file d'attente

Cet exemple permet au AWS compte spécifié d'envoyer des messages à la file d'attente spécifiée.

Commande :

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/
MyQueue --label SendMessagesFromMyQueue --aws-account-ids 12345EXAMPLE --
actions SendMessage
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir AddPermissionla section Référence des AWS CLI commandes.

`cancel-message-move-task`

L'exemple de code suivant montre comment utiliser`cancel-message-move-task`.

AWS CLI

Pour annuler une tâche de déplacement de message

L'`cancel-message-move-task`exemple suivant annule la tâche de déplacement de message spécifiée.

```
aws sqs cancel-message-move-task \
    --task-handle AQEB6nR4...HzlvZQ==
```

Sortie :

```
{
    "ApproximateNumberOfMessagesMoved": 102
}
```

Pour plus d'informations, consultez la section Autorisations d'API Amazon SQS : références relatives aux actions et aux ressources dans le manuel du développeur.

- Pour plus de détails sur l'API, voir CancelMessageMoveTaskla section Référence des AWS CLI commandes.

`change-message-visibility-batch`

L'exemple de code suivant montre comment utiliser`change-message-visibility-batch`.

AWS CLI

Pour modifier la visibilité du délai d'expiration de plusieurs messages sous forme de lot

Cet exemple fait passer le délai d'expiration des 2 messages spécifiés à 10 heures (10 heures* 60 minutes* 60 secondes).

Commande :

```
aws sqs change-message-visibility-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://change-message-
visibility-batch.json
```

Fichier d'entrée (change-message-visibility-batch.json) :

```
[
  {
    "Id": "FirstMessage",
        "ReceiptHandle": "AQEBhz2q...Jf3kaw==",
        "VisibilityTimeout": 36000
  },
  {
    "Id": "SecondMessage",
        "ReceiptHandle": "AQEBkTUH...HifSnw==",
        "VisibilityTimeout": 36000
  }
]
```

Sortie :

```
{
  "Successful": [
    {
      "Id": "SecondMessage"
    },
    {
      "Id": "FirstMessage"
    }
  ]
}
```

- Pour plus de détails sur l'API, voir ChangeMessageVisibilityBatchla section Référence des AWS CLI commandes.

`change-message-visibility`

L'exemple de code suivant montre comment utiliser`change-message-visibility`.

AWS CLI

Pour modifier la visibilité du délai d'expiration d'un message

Cet exemple fait passer le délai de visibilité du message spécifié à 10 heures (10 heures* 60 minutes* 60 secondes).

Commande :

```
aws sqs change-message-visibility --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --
visibility-timeout 36000
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir ChangeMessageVisibilityla section Référence des AWS CLI commandes.

`create-queue`

L'exemple de code suivant montre comment utiliser`create-queue`.

AWS CLI

Pour créer une file d'attente

Cet exemple crée une file d'attente portant le nom spécifié, définit la période de rétention des messages à 3 jours (3 jours* 24 heures* 60 minutes* 60 secondes) et définit la file d'attente de lettres mortes de la file d'attente comme étant la file d'attente spécifiée avec un nombre maximum de 1 000 messages reçus.

Commande :

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

Fichier d'entrée (create-queue.json) :

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "MessageRetentionPeriod": "259200"
}
```

Sortie :

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- Pour plus de détails sur l'API, voir CreateQueuela section Référence des AWS CLI commandes.

`delete-message-batch`

L'exemple de code suivant montre comment utiliser`delete-message-batch`.

AWS CLI

Pour supprimer plusieurs messages par lots

Cet exemple supprime les messages spécifiés.

Commande :

```
aws sqs delete-message-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

Fichier d'entrée (delete-message-batch.json) :

```
[
  {
        "Id": "FirstMessage",
        "ReceiptHandle": "AQEB1mgl...Z4GuLw=="
  },
  {
    "Id": "SecondMessage",
        "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

Sortie :

```
{
  "Successful": [
    {
```

```
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

- Pour plus de détails sur l'API, voir DeleteMessageBatchla section Référence des AWS CLI commandes.

`delete-message`

L'exemple de code suivant montre comment utiliser`delete-message`.

AWS CLI

Pour supprimer un message

Cet exemple supprime le message spécifié.

Commande :

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir DeleteMessagela section Référence des AWS CLI commandes.

`delete-queue`

L'exemple de code suivant montre comment utiliser`delete-queue`.

AWS CLI

Pour supprimer une file d'attente

Cet exemple supprime la file d'attente spécifiée.

Commande :

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir DeleteQueuela section Référence des AWS CLI commandes.

`get-queue-attributes`

L'exemple de code suivant montre comment utiliser`get-queue-attributes`.

AWS CLI

Pour obtenir les attributs d'une file d'attente

Cet exemple obtient tous les attributs de la file d'attente spécifiée.

Commande :

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All
```

Sortie :

```
{
  "Attributes": {
    "ApproximateNumberOfMessagesNotVisible": "0",
    "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":1000}",
    "MessageRetentionPeriod": "345600",
    "ApproximateNumberOfMessagesDelayed": "0",
    "MaximumMessageSize": "262144",
    "CreatedTimestamp": "1442426968",
    "ApproximateNumberOfMessages": "0",
```

```
    "ReceiveMessageWaitTimeSeconds": "0",
    "DelaySeconds": "0",
    "VisibilityTimeout": "30",
    "LastModifiedTimestamp": "1442426968",
    "QueueArn": "arn:aws:sqs:us-east-1:80398EXAMPLE:MyNewQueue"
  }
}
```

Cet exemple obtient uniquement la taille maximale des messages et les attributs de délai de visibilité de la file d'attente spécifiée.

Commande :

```
aws sqs get-queue-attributes --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-
names MaximumMessageSize VisibilityTimeout
```

Sortie :

```
{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}
```

- Pour plus de détails sur l'API, voir GetQueueAttributesla section Référence des AWS CLI commandes.

`get-queue-url`

L'exemple de code suivant montre comment utiliser`get-queue-url`.

AWS CLI

Pour obtenir l'URL d'une file d'attente

Cet exemple permet d'obtenir l'URL de la file d'attente spécifiée.

Commande :

```
aws sqs get-queue-url --queue-name MyQueue
```

Sortie :

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- Pour plus de détails sur l'API, voir GetQueueUrlla section Référence des AWS CLI commandes.

`list-dead-letter-source-queues`

L'exemple de code suivant montre comment utiliser`list-dead-letter-source-queues`.

AWS CLI

Pour répertorier les files d'attente contenant des sources de lettres mortes

Cet exemple répertorie les files d'attente associées à la file source de lettres mortes spécifiée.

Commande :

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

Sortie :

```
{
  "queueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- Pour plus de détails sur l'API, voir ListDeadLetterSourceQueuesla section Référence des AWS CLI commandes.

`list-message-move-tasks`

L'exemple de code suivant montre comment utiliser`list-message-move-tasks`.

AWS CLI

Pour répertorier le message, déplacez les tâches

L'`list-message-move-tasks`exemple suivant répertorie les deux tâches de déplacement de messages les plus récentes dans la file d'attente spécifiée.

```
aws sqs list-message-move-tasks \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \
    --max-results 2
```

Sortie :

```
{
    "Results": [
        {
            "TaskHandle": "AQEB6nR4...HzlvZQ==",
            "Status": "RUNNING",
            "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",
            "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",
            "MaxNumberOfMessagesPerSecond": 50,
            "ApproximateNumberOfMessagesMoved": 203,
            "ApproximateNumberOfMessagesToMove": 30,
            "StartedTimestamp": 1442428276921
         },

         {
            "Status": "COMPLETED",
            "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",
            "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",
            "ApproximateNumberOfMessagesMoved": 29,
            "ApproximateNumberOfMessagesToMove": 0,
            "StartedTimestamp": 1342428272093
         }
    ]
}
```

Pour plus d'informations, consultez la section Autorisations d'API Amazon SQS : références relatives aux actions et aux ressources dans le manuel du développeur.

- Pour plus de détails sur l'API, voir ListMessageMoveTasksla section Référence des AWS CLI commandes.

`list-queue-tags`

L'exemple de code suivant montre comment utiliser`list-queue-tags`.

AWS CLI

Pour répertorier toutes les balises de répartition des coûts pour une file d'attente

L'`list-queue-tags`exemple suivant affiche toutes les balises de répartition des coûts associées à la file d'attente spécifiée.

```
aws sqs list-queue-tags \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue
```

Sortie :

```
{
    "Tags": {
        "Team": "Alpha"
    }
}
```

Pour plus d'informations, consultez Listing Cost Allocation Tags dans le manuel Amazon Simple Queue Service Developer Guide.

- Pour plus de détails sur l'API, voir ListQueueTagsla section Référence des AWS CLI commandes.

`list-queues`

L'exemple de code suivant montre comment utiliser`list-queues`.

AWS CLI

Pour répertorier les files d'attente

Cet exemple répertorie toutes les files d'attente.

Commande :

```
aws sqs list-queues
```

Sortie :

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/TestQueue1",
        "https://queue.amazonaws.com/80398EXAMPLE/TestQueue2"
  ]
}
```

Cet exemple répertorie uniquement les files d'attente qui commencent par « Mon ».

Commande :

```
aws sqs list-queues --queue-name-prefix My
```

Sortie :

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- Pour plus de détails sur l'API, voir ListQueuesla section Référence des AWS CLI commandes.

`purge-queue`

L'exemple de code suivant montre comment utiliser`purge-queue`.

AWS CLI

Pour purger une file d'attente

Cet exemple supprime tous les messages de la file d'attente spécifiée.

Commande :

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir PurgeQueuela section Référence des AWS CLI commandes.

`receive-message`

L'exemple de code suivant montre comment utiliser`receive-message`.

AWS CLI

Pour recevoir un message

Cet exemple reçoit jusqu'à 10 messages disponibles et renvoie tous les attributs disponibles.

Commande :

```
aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All --message-attribute-names All --max-number-of-messages 10
```

Sortie :

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5OfBody": "1000f835...a35411fa",
      "MD5OfMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "ApproximateReceiveCount": "5",
        "SentTimestamp": "1442428276921"
```

```
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        },
        "City": {
          "DataType": "String",
          "StringValue": "Any City"
        }
      }
    }
  ]
}
```

Cet exemple reçoit le prochain message disponible, renvoyant uniquement les SentTimestamp attributs SenderId et ainsi que l'attribut PostalCode message.

Commande :

```
aws sqs receive-message --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp
 --message-attribute-names PostalCode
```

Sortie :

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5OfBody": "1000f835...a35411fa",
      "MD5OfMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        }
```

```
      }
    }
  ]
}
```

- Pour plus de détails sur l'API, voir ReceiveMessagela section Référence des AWS CLI commandes.

`remove-permission`

L'exemple de code suivant montre comment utiliser`remove-permission`.

AWS CLI

Pour supprimer une autorisation

Cet exemple supprime l'autorisation portant l'étiquette spécifiée de la file d'attente spécifiée.

Commande :

```
aws sqs remove-permission --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessagesFromMyQueue
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir RemovePermissionla section Référence des AWS CLI commandes.

`send-message-batch`

L'exemple de code suivant montre comment utiliser`send-message-batch`.

AWS CLI

Pour envoyer plusieurs messages par lots

Cet exemple envoie 2 messages avec les corps de message, les délais et les attributs de message spécifiés à la file d'attente spécifiée.

Commande :

```
aws sqs send-message-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

Fichier d'entrée (send-message-batch.json) :

```
[
  {
    "Id": "FuelReport-0001-2015-09-16T140731Z",
        "MessageBody": "Fuel report for account 0001 on 2015-09-16 at 02:07:31 PM.",
        "DelaySeconds": 10,
        "MessageAttributes": {
          "SellerName": {
            "DataType": "String",
                "StringValue": "Example Store"
     },
          "City": {
        "DataType": "String",
        "StringValue": "Any City"
     },
          "Region": {
            "DataType": "String",
                "StringValue": "WA"
     },
          "PostalCode": {
            "DataType": "String",
                "StringValue": "99065"
          },
          "PricePerGallon": {
            "DataType": "Number",
                "StringValue": "1.99"
     }
        }
  },
  {
    "Id": "FuelReport-0002-2015-09-16T140930Z",
        "MessageBody": "Fuel report for account 0002 on 2015-09-16 at 02:09:30 PM.",
        "DelaySeconds": 10,
        "MessageAttributes": {
          "SellerName": {
            "DataType": "String",
                "StringValue": "Example Fuels"
```

```
      },
          "City": {
        "DataType": "String",
        "StringValue": "North Town"
      },
          "Region": {
            "DataType": "String",
                "StringValue": "WA"
      },
          "PostalCode": {
            "DataType": "String",
                "StringValue": "99123"
          },
          "PricePerGallon": {
            "DataType": "Number",
                "StringValue": "1.87"
      }
        }
  }
]
```

Sortie :

```
{
  "Successful": [
    {
      "MD5OfMessageBody": "203c4a38...7943237e",
      "MD5OfMessageAttributes": "10809b55...baf283ef",
      "Id": "FuelReport-0001-2015-09-16T140731Z",
      "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
    },
    {
      "MD5OfMessageBody": "2cf0159a...c1980595",
      "MD5OfMessageAttributes": "55623928...ae354a25",
      "Id": "FuelReport-0002-2015-09-16T140930Z",
      "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
    }
  ]
}
```

- Pour plus de détails sur l'API, voir SendMessageBatchla section Référence des AWS CLI commandes.

`send-message`

L'exemple de code suivant montre comment utiliser`send-message`.

AWS CLI

Pour envoyer un message

Cet exemple envoie un message avec le corps du message, le délai et les attributs de message spécifiés à la file d'attente spécifiée.

Commande :

```
aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --message-body "Information about the largest city in Any Region." --delay-seconds 10 --message-attributes file://send-message.json
```

Fichier d'entrée (send-message.json) :

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
    "DataType": "Binary",
    "BinaryValue": "Hello, World!"
  },
  "Population": {
    "DataType": "Number",
    "StringValue": "1250800"
  }
}
```

Sortie :

```
{
  "MD5OfMessageBody": "51b0a325...39163aa0",
  "MD5OfMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```

- Pour plus de détails sur l'API, voir SendMessagela section Référence des AWS CLI commandes.

`set-queue-attributes`

L'exemple de code suivant montre comment utiliser`set-queue-attributes`.

AWS CLI

Pour définir les attributs de file d'attente

Cet exemple définit pour la file d'attente spécifiée un délai de livraison de 10 secondes, une taille de message maximale de 128 Ko (128 Ko x 1 024 octets), une période de rétention des messages de 3 jours (3 jours* 24 heures* 60 minutes* 60 secondes), un temps d'attente des messages de réception de 20 secondes et un délai de visibilité par défaut de 60 secondes. Cet exemple associe également la file d'attente de lettres mortes spécifiée à un nombre maximum de 1 000 messages reçus.

Commande :

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

Fichier d'entrée (set-queue-attributes.json) :

```
{
  "DelaySeconds": "10",
  "MaximumMessageSize": "131072",
  "MessageRetentionPeriod": "259200",
  "ReceiveMessageWaitTimeSeconds": "20",
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "VisibilityTimeout": "60"
}
```

Sortie :

```
None.
```

- Pour plus de détails sur l'API, voir SetQueueAttributesla section Référence des AWS CLI commandes.

`start-message-move-task`

L'exemple de code suivant montre comment utiliser`start-message-move-task`.

AWS CLI

Exemple 1 : *Pour démarrer une tâche de déplacement de messages*

L'`start-message-move-task`exemple suivant lance une tâche de déplacement de messages pour rediriger les messages de la file d'attente de lettres mortes spécifiée vers la file d'attente source.

```
aws sqs start-message-move-task \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

Sortie :

```
{
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

Pour plus d'informations, voir Ceci est le titre de la rubrique dans le nom de votre guide.

Exemple 2 : *Pour démarrer une tâche de déplacement de messages avec un taux maximal*

L'`start-message-move-task`exemple suivant lance une tâche de déplacement de messages pour rediriger les messages de la file d'attente de lettres mortes spécifiée vers la file de destination spécifiée à une cadence maximale de 50 messages par seconde.

```
aws sqs start-message-move-task \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1 \
    --destination-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2 \
    --max-number-of-messages-per-second 50
```

Sortie :

```
{
```

```
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

Pour plus d'informations, consultez la section Autorisations d'API Amazon SQS : références relatives aux actions et aux ressources dans le manuel du développeur.

- Pour plus de détails sur l'API, voir StartMessageMoveTaskla section Référence des AWS CLI commandes.

`tag-queue`

L'exemple de code suivant montre comment utiliser`tag-queue`.

AWS CLI

Pour ajouter des balises de répartition des coûts à une file d'attente

L'`tag-queue`exemple suivant ajoute une balise de répartition des coûts à la file d'attente Amazon SQS spécifiée.

```
aws sqs tag-queue \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \
    --tags Priority=Highest
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout de balises de répartition des coûts dans le manuel Amazon Simple Queue Service Developer Guide.

- Pour plus de détails sur l'API, voir TagQueuela section Référence des AWS CLI commandes.

`untag-queue`

L'exemple de code suivant montre comment utiliser`untag-queue`.

AWS CLI

Pour supprimer les balises de répartition des coûts d'une file d'attente

L'`untag-queue`exemple suivant supprime une balise de répartition des coûts de la file d'attente Amazon SQS spécifiée.

```
aws sqs untag-queue \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \
    --tag-keys "Priority"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Ajout de balises de répartition des coûts dans le manuel Amazon Simple Queue Service Developer Guide.

- Pour plus de détails sur l'API, voir UntagQueuela section Référence des AWS CLI commandes.

Exemples de Storage Gateway utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Storage Gateway.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`describe-gateway-information`

L'exemple de code suivant montre comment utiliser`describe-gateway-information`.

AWS CLI

Pour décrire une passerelle

La `describe-gateway-information` commande suivante renvoie les métadonnées relatives à la passerelle spécifiée. Pour spécifier la passerelle à décrire, utilisez l'Amazon Resource Name (ARN) de la passerelle dans la commande.

Cet exemple spécifie une passerelle avec l'identifiant `sgw-12A3456B` dans le compte `123456789012` :

```
aws storagegateway describe-gateway-information --gateway-
arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Cette commande génère un bloc JSON contenant des métadonnées relatives à la passerelle, telles que son nom, ses interfaces réseau, son fuseau horaire configuré et son état (que la passerelle soit en cours d'exécution ou non).

- Pour plus de détails sur l'API, reportez-vous DescribeGatewayInformationà la section Référence des AWS CLI commandes.

`list-file-shares`

L'exemple de code suivant montre comment utiliser`list-file-shares`.

AWS CLI

Pour répertorier les partages de fichiers

L'`command-name`exemple suivant répertorie les widgets disponibles dans votre AWS compte.

```
aws storagegateway list-file-shares \
    --gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

Sortie :

```
{
    "FileShareInfoList": [
        {
            "FileShareType": "NFS",
            "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345",
            "FileShareId": "share-2FA12345",
            "FileShareStatus": "AVAILABLE",
            "GatewayARN": "arn:aws:storagegateway:us-east-1:111122223333:gateway/
sgw-FB0AAAAA"
        }
    ],
    "Marker": null
```

```
}
```

Pour plus d'informations, consultez ListFileSharesle manuel AWS Storage Gateway Service API Reference.

- Pour plus de détails sur l'API, reportez-vous ListFileSharesà la section Référence des AWS CLI commandes.

`list-gateways`

L'exemple de code suivant montre comment utiliser`list-gateways`.

AWS CLI

Pour répertorier les passerelles d'un compte

La `list-gateways` commande suivante répertorie toutes les passerelles définies pour un compte :

```
aws storagegateway list-gateways
```

Cette commande génère un bloc JSON qui contient une liste des noms de ressources Amazon (ARNs) de la passerelle.

- Pour plus de détails sur l'API, reportez-vous ListGatewaysà la section Référence des AWS CLI commandes.

`list-volumes`

L'exemple de code suivant montre comment utiliser`list-volumes`.

AWS CLI

Pour répertorier les volumes configurés pour une passerelle

La `list-volumes` commande suivante renvoie la liste des volumes configurés pour la passerelle spécifiée. Pour spécifier la passerelle à décrire, utilisez l'Amazon Resource Name (ARN) de la passerelle dans la commande.

Cet exemple spécifie une passerelle avec l'identifiant `sgw-12A3456B` dans le compte `123456789012` :

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-
west-2:123456789012:gateway/sgw-12A3456B"
```

Cette commande génère un bloc JSON contenant une liste de volumes incluant le type et l'ARN de chaque volume.

- Pour plus de détails sur l'API, reportez-vous ListVolumesà la section Référence des AWS CLI commandes.

`refresh-cache`

L'exemple de code suivant montre comment utiliser`refresh-cache`.

AWS CLI

Pour actualiser le cache de partage de fichiers

L'`refresh-cache`exemple suivant actualise le cache pour le partage de fichiers spécifié.

```
aws storagegateway refresh-cache \
    --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345
```

Sortie :

```
{
    "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345",
    "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"
}
```

Pour plus d'informations, consultez ListFileSharesle manuel AWS Storage Gateway Service API Reference.

- Pour plus de détails sur l'API, reportez-vous RefreshCacheà la section Référence des AWS CLI commandes.

AWS STS exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS STS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`assume-role-with-saml`

L'exemple de code suivant montre comment utiliser`assume-role-with-saml`.

AWS CLI

Pour obtenir des informations d'identification à court terme pour un rôle authentifié avec SAML

La commande `assume-role-with-saml` suivante extrait un ensemble d'informations d'identification à court terme pour le rôle IAM `TestSaml`. Dans cet exemple, la demande est authentifiée à l'aide de l'assertion SAML fournie par votre fournisseur d'identité lorsque vous vous authentifiez auprès de celui-ci.

```
aws sts assume-role-with-saml \
    --role-arn arn:aws:iam::123456789012:role/TestSaml \
    --principal-arn arn:aws:iam::123456789012:saml-provider/SAML-test \
    --saml-
assertion "VERYLONGENCODEDASSERTIONEXAMPLExzYW1sOkF1ZGllbmNlPmJsYW5rPC9zYW1sOkF1ZGllbmNlPjwv
+PHNhbWw6TmFtZUlEIEZvcm1hdD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOm5hbWVpZC1mb3JtYXQ6dHJhbnN
+PHNhbWw6U3ViamVjdENvbmZpcm1hdGlvbiBNZXRob2Q9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpjbTpiZWF
```

Sortie :

```
{
    "Issuer": "https://integ.example.com/idp/shibboleth</Issuer",
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/TestSaml",
        "AssumedRoleId": "ARO456EXAMPLE789:TestSaml"
    },
```

```
    "Credentials": {
        "AccessKeyId": "ASIAV3ZUEFP6EXAMPLE",
        "SecretAccessKey": "8P+SQvWIuLnKhh8d++jpw0nNmQRBZvNEXAMPLEKEY",
        "SessionToken": "IQoJb3JpZ2luX2VjEOz////////////////////
wEXAMPLEtMSJHMEUCIDoKK3JH9uGQE1z0sINr5M4jk
+Na8KHDcCYRVjJCZEvOAiEA3OvJGtw1EcViOleS2vhs8VdCKFJQWPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==",
        "Expiration": "2019-11-01T20:26:47Z"
    },
    "Audience": "https://signin.aws.amazon.com/saml",
    "SubjectType": "transient",
    "PackedPolicySize": "6",
    "NameQualifier": "SbdGOnUkh1i4+EXAMPLExL/jEvs=",
    "Subject": "SamlExample"
}
```

Pour plus d'informations, consultez Demande d'informations d'identification temporaires de sécurité dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AssumeRoleWithSamlà la section Référence des AWS CLI commandes.

`assume-role-with-web-identity`

L'exemple de code suivant montre comment utiliser`assume-role-with-web-identity`.

AWS CLI

Pour obtenir des informations d'identification à court terme pour un rôle authentifié avec Web Identity (OAuth 2."0)

La commande `assume-role-with-web-identity` suivante extrait un ensemble d'informations d'identification à court terme pour le rôle IAM `app1`. La requête est authentifiée à l'aide du jeton d'identité Web fourni par le fournisseur d'identité Web spécifié. Deux politiques supplémentaires sont appliquées à la session afin de restreindre davantage ce que l'utilisateur peut faire. Les informations d'identification renvoyées expirent une heure après avoir été générées.

```
aws sts assume-role-with-web-identity \
    --duration-seconds 3600 \
    --role-session-name "app1" \
```

```
    --provider-id "www.amazon.com" \
    --policy-arns "arn:aws:iam::123456789012:policy/
q=webidentitydemopolicy1","arn:aws:iam::123456789012:policy/webidentitydemopolicy2"
 \
    --role-arn arn:aws:iam::123456789012:role/FederatedWebIdentityRole \
    --web-identity-token "Atza
%7CIQEBLjAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXJnrulxKDHwy87oGKPznh0D6bEQZTSCzyoC
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wl7WTI7jn-Pcb6M-
buCgHhFOzTQxod27L9CqnOLio7N3gZAGpsp6n1-
AJBOCJckcyXe2c6uD0srOJeZlKUm2eTDVMf8IehDVI0r1QOnTV6KzzAI3OY87Vd_cVMQ"
```

Sortie :

```
{
    "SubjectFromWebIdentityToken": "amzn1.account.AF6RHO7KZU5XRVQJGXK6HB56KR2A",
    "Audience": "client.5498841531868486423.1548@apps.example.com",
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/
app1",
        "AssumedRoleId": "AROACLKWSDQRAOEXAMPLE:app1"
    },
    "Credentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    },
    "Provider": "www.amazon.com"
}
```

Pour plus d'informations, consultez Demande d'informations d'identification temporaires de sécurité dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AssumeRoleWithWebIdentityà la section Référence des AWS CLI commandes.

`assume-role`

L'exemple de code suivant montre comment utiliser`assume-role`.

AWS CLI

Pour endosser un rôle

La commande `assume-role` suivante extrait un ensemble d'informations d'identification à court terme pour le rôle IAM `s3-access-example`.

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

Sortie :

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-
example"
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

La sortie de la commande contient une clé d'accès, une clé secrète et un jeton de session que vous pouvez utiliser pour vous authentifier auprès d' AWS.

Pour l'utilisation de la AWS CLI, vous pouvez configurer un profil nommé associé à un rôle. Lorsque vous utilisez le profil, la AWS CLI appelle assume-role et gère les informations d'identification pour vous. Pour plus d'informations, consultez la section Utiliser un rôle IAM dans la AWS CLI dans le Guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous AssumeRoleà la section Référence des AWS CLI commandes.

`assume-root`

L'exemple de code suivant montre comment utiliser`assume-root`.

AWS CLI

Pour lancer une session privilégiée

La `assume-root` commande suivante permet de récupérer un ensemble d'informations d'identification à court terme que vous pouvez utiliser pour supprimer une politique de compartiment Amazon S3 mal configurée pour un compte membre de votre organisation.

```
aws sts assume-root \
    --duration-seconds 900 \
    --target-principal 111122223333 \
    --task-policy-arn arn=arn:aws:iam::aws:policy/root-task/S3UnlockBucketPolicy
```

Sortie :

```
{
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2024-11-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    },
    "SourceIdentity": "Alice",
}
```

La sortie de la commande contient une clé d'accès, une clé secrète et un jeton de session que vous pouvez utiliser pour effectuer des actions privilégiées sur le compte membre. Pour plus d'informations, voir Exécuter une tâche privilégiée sur le compte d'un membre AWS d'une Organisation dans le Guide de l'utilisateur d'AWS IAM.

- Pour plus de détails sur l'API, reportez-vous AssumeRootà la section Référence des AWS CLI commandes.

`decode-authorization-message`

L'exemple de code suivant montre comment utiliser`decode-authorization-message`.

AWS CLI

Pour décoder un message d'autorisation codé renvoyé en réponse à une requête

L'exemple `decode-authorization-message` suivant décode des informations supplémentaires sur le statut d'autorisation d'une demande émise depuis un message codé envoyé en réponse à une requête Amazon Web Services.

```
aws sts decode-authorization-message \
    --encoded-message EXAMPLEWodyRNrtlQARDip-
eTA6i6DrlUhHhPQrLWB_lAbl5pAKxl9mPDLexYcGBreyIKQC1BGBIpBKr3dFDkwqeO7e2NMk5j_hmzAiChJN-8oy3Ewi
Ojau7BMjOTWwOtHPhV_Zaz87yENdipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK44E
pOu3FZjwYStfvTb3GHs3-6rLribGO9jZOktkfE6vqxlFzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVWPxioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHlI-
tIiNd7tgNTYzDuPQYD2FMDBnp82V9eVmYGtPp5NIeSpuf3fOHanFuBZgENxZQZ2dlH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067POFrZECLjjOtNPBOZCcuEKEXAM
```

Sortie :

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements
\":{\"items\":[{\"statementId\":\"VisualEditor0\",\"effect\":\"DENY\",\"principals
\":{\"items\":[{\"value\":\"AROA123456789EXAMPLE\"}]},\"principalGroups
\":{\"items\":[]},\"actions\":{\"items\":[{\"value\":\"ec2:RunInstances
\"}]},\"resources\":{\"items\":[{\"value\":\"*\"}]},\"conditions\":{\"items
\":[]}}]},\"failures\":{\"items\":[]},\"context\":{\"principal\":{\"id\":
\"AROA123456789EXAMPLE:Ana\",\"arn\":\"arn:aws:sts::111122223333:assumed-role/
Developer/Ana\"},\"action\":\"RunInstances\",\"resource\":\"arn:aws:ec2:us-
east-1:111122223333:instance/*\",\"conditions\":{\"items\":[{\"key\":
\"ec2:MetadataHttpPutResponseHopLimit\",\"values\":{\"items\":[{\"value\":
\"2\"}]}},{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value
\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value
\":\"instance/*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value
\":\"111122223333\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":
[{\"value\":\"us-east-1f\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items
\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values
\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values\":
{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",\"values
```

```
\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values\":{\"items
\":[{\"value\":\"us-east-1\"}]}},{\"key\":\"ec2:MetadataHttpEndpoint\",\"values
\":{\"items\":[{\"value\":\"enabled\"}]}},{\"key\":\"aws:Service\",\"values\":
{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":{\"items
\":[{\"value\":\"*\"}]}},{\"key\":\"ec2:MetadataHttpTokens\",\"values\":{\"items
\":[{\"value\":\"required\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":
[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":
[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-1\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":
\"arn:aws:ec2:us-east-1:111122223333:instance/*\"}]}}]}}}"
}
```

Pour de plus amples informations, consultez Logique d'évaluation des politiques dans le AWS Guide de l'utilisateur IAM.

- Pour plus de détails sur l'API, reportez-vous DecodeAuthorizationMessageà la section Référence des AWS CLI commandes.

`get-caller-identity`

L'exemple de code suivant montre comment utiliser`get-caller-identity`.

AWS CLI

Pour obtenir des informations sur l'identité IAM actuelle

La `get-caller-identity` commande suivante affiche des informations sur l'identité IAM utilisée pour authentifier la demande. L'appelant est un utilisateur IAM.

```
aws sts get-caller-identity
```

Sortie :

```
{
    "UserId": "AIDASAMPLEUSERID",
    "Account": "123456789012",
    "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- Pour plus de détails sur l'API, reportez-vous GetCallerIdentityà la section Référence des AWS CLI commandes.

`get-federation-token`

L'exemple de code suivant montre comment utiliser`get-federation-token`.

AWS CLI

Pour renvoyer un ensemble d'informations d'identification de sécurité temporaires à l'aide des informations d'identification de la clé d'accès utilisateur IAM

L'exemple `get-federation-token` suivant renvoie un ensemble d'informations d'identification de sécurité temporaires (composé d'un ID de clé d'accès, d'une clé d'accès secrète et d'un jeton de sécurité) pour un utilisateur. Vous devez appeler l'opération `GetFederationToken` à l'aide des informations d'identification de sécurité à long terme d'un utilisateur IAM.

```
aws sts get-federation-token \
    --name Bob \
    --policy file://myfile.json \
    --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
    --duration-seconds 900
```

Contenu de `myfile.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:Describe*"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:Describe*",
            "Resource": "*"
        }
    ]
}
```

Sortie :

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "EXAMPLEpZ2luX2VjEGoaCXVzLXdlc3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QQunWMTfKq0DCOP/////////
wEQAxoMNDUyOTI1MTcwNTA3Igxuy3AOpuuoLsk3MJwqgQPg8QOd9HuoClUxq26wnc/nm
+eZLjHDyGf2KUAHK2DuaS/nrGSEXAMPLE",
        "Expiration": "2023-12-20T02:06:07+00:00"
    },
    "FederatedUser": {
        "FederatedUserId": "111122223333:Bob",
        "Arn": "arn:aws:sts::111122223333:federated-user/Bob"
    },
    "PackedPolicySize": 36
}
```

Pour plus d'informations, consultez Demande d'informations d'identification temporaires de sécurité dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetFederationTokenà la section Référence des AWS CLI commandes.

`get-session-token`

L'exemple de code suivant montre comment utiliser`get-session-token`.

AWS CLI

Pour obtenir un ensemble d'informations d'identification à court terme d'une identité IAM

La commande `get-session-token` suivante extrait un ensemble d'informations d'identification à court terme pour l'identité IAM qui effectue l'appel. Les informations d'identification obtenues peuvent être utilisées pour les requêtes où l'authentification multifactorielle (MFA) est requise par la politique. Les informations d'identification expirent 15 minutes après leur création.

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
    --token-code 123456
```

Sortie :

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

Pour plus d'informations, consultez Demande d'informations d'identification temporaires de sécurité dans le Guide de l'utilisateur AWS IAM.

- Pour plus de détails sur l'API, reportez-vous GetSessionTokenà la section Référence des AWS CLI commandes.

Support exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with Support.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-attachments-to-set`

L'exemple de code suivant montre comment utiliser`add-attachments-to-set`.

AWS CLI

Pour ajouter une pièce jointe à un ensemble

L'`add-attachments-to-set`exemple suivant ajoute une image à un ensemble que vous pouvez ensuite spécifier pour un dossier d'assistance dans votre AWS compte.

```
aws support add-attachments-to-set \
    --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE" \
    --attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

Sortie :

```
{
    "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE",
    "expiryTime": "2020-05-14T17:04:40.790+0000"
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir AddAttachmentsToSetla section Référence des AWS CLI commandes.

`add-communication-to-case`

L'exemple de code suivant montre comment utiliser`add-communication-to-case`.

AWS CLI

Pour ajouter une communication à un dossier

L'`add-communication-to-case`exemple suivant ajoute des communications à une demande d'assistance dans votre AWS compte.

```
aws support add-communication-to-case \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
    --communication-body "I'm attaching a set of images to this case." \
    --cc-email-addresses "myemail@example.com" \
    --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE"
```

Sortie :

```
{
    "result": true
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir AddCommunicationToCasela section Référence des AWS CLI commandes.

`create-case`

L'exemple de code suivant montre comment utiliser`create-case`.

AWS CLI

Pour créer un dossier

L'`create-case`exemple suivant crée une demande d'assistance pour votre AWS compte.

```
aws support create-case \
    --category-code "using-aws" \
    --cc-email-addresses "myemail@example.com" \
    --communication-body "I want to learn more about an AWS service." \
    --issue-type "technical" \
```

```
    --language "en" \
    --service-code "general-info" \
    --severity-code "low" \
    --subject "Question about my account"
```

Sortie :

```
{
    "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir CreateCasela section Référence des AWS CLI commandes.

`describe-attachment`

L'exemple de code suivant montre comment utiliser`describe-attachment`.

AWS CLI

Pour décrire une pièce jointe

L'`describe-attachment`exemple suivant renvoie des informations sur la pièce jointe avec l'ID spécifié.

```
aws support describe-attachment \
    --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakqlc60-
iJjL5HqyYGiT1FG8EXAMPLE"
```

Sortie :

```
{
    "attachment": {
        "fileName": "troubleshoot-screenshot.png",
        "data": "base64-blob"
    }
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeAttachmentla section Référence des AWS CLI commandes.

`describe-cases`

L'exemple de code suivant montre comment utiliser`describe-cases`.

AWS CLI

Pour décrire un cas

L'`describe-cases`exemple suivant renvoie des informations sur le dossier d'assistance spécifié dans votre AWS compte.

```
aws support describe-cases \
    --display-id "1234567890" \
    --after-time "2020-03-23T21:31:47.774Z" \
    --include-resolved-cases \
    --language "en" \
    --no-include-communications \
    --max-item 1
```

Sortie :

```
{
    "cases": [
        {
            "status": "resolved",
            "ccEmailAddresses": [],
            "timeCreated": "2020-03-23T21:31:47.774Z",
            "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
            "severityCode": "low",
            "language": "en",
            "categoryCode": "using-aws",
            "serviceCode": "general-info",
            "submittedBy": "myemail@example.com",
            "displayId": "1234567890",
            "subject": "Question about my account"
        }
```

```
    ]
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeCaseslа section Référence des AWS CLI commandes.

`describe-communications`

L'exemple de code suivant montre comment utiliser`describe-communications`.

AWS CLI

Pour décrire la dernière communication concernant un dossier

L'`describe-communications`exemple suivant renvoie la dernière communication concernant le dossier d'assistance spécifié dans votre AWS compte.

```
aws support describe-communications \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
    --after-time "2020-03-23T21:31:47.774Z" \
    --max-item 1
```

Sortie :

```
{
    "communications": [
        {
            "body": "I want to learn more about an AWS service.",
            "attachmentSet": [],
            "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
            "timeCreated": "2020-05-12T23:12:35.000Z",
            "submittedBy": "Amazon Web Services"
        }
    ],
    "NextToken": "eyJuZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeCommunicationsla section Référence des AWS CLI commandes.

`describe-services`

L'exemple de code suivant montre comment utiliser`describe-services`.

AWS CLI

Pour répertorier AWS les services et les catégories de services

L'`describe-services`exemple suivant répertorie les catégories de services disponibles pour demander des informations générales.

```
aws support describe-services \
    --service-code-list "general-info"
```

Sortie :

```
{
    "services": [
        {
            "code": "general-info",
            "name": "General Info and Getting Started",
            "categories": [
                {
                    "code": "charges",
                    "name": "How Will I Be Charged?"
                },
                {
                    "code": "gdpr-queries",
                    "name": "Data Privacy Query"
                },
                {
                    "code": "reserved-instances",
                    "name": "Reserved Instances"
                },
                {
                    "code": "resource",
                    "name": "Where is my Resource?"
                },
                {
```

```
                    "code": "using-aws",
                    "name": "Using AWS & Services"
                },
                {
                    "code": "free-tier",
                    "name": "Free Tier"
                },
                {
                    "code": "security-and-compliance",
                    "name": "Security & Compliance"
                },
                {
                    "code": "account-structure",
                    "name": "Account Structure"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeServicesla section Référence des AWS CLI commandes.

`describe-severity-levels`

L'exemple de code suivant montre comment utiliser`describe-severity-levels`.

AWS CLI

Pour répertorier les niveaux de gravité disponibles

L'`describe-severity-levels`exemple suivant répertorie les niveaux de gravité disponibles pour un dossier de support.

```
aws support describe-severity-levels
```

Sortie :

```
{
```

```
    "severityLevels": [
        {
            "code": "low",
            "name": "Low"
        },
        {
            "code": "normal",
            "name": "Normal"
        },
        {
            "code": "high",
            "name": "High"
        },
        {
            "code": "urgent",
            "name": "Urgent"
        },
        {
            "code": "critical",
            "name": "Critical"
        }
    ]
}
```

Pour plus d'informations, consultez la section Choix d'une sévérité dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeSeverityLevelsla section Référence des AWS CLI commandes.

`describe-trusted-advisor-check-refresh-statuses`

L'exemple de code suivant montre comment utiliser`describe-trusted-advisor-check-refresh-statuses`.

AWS CLI

Pour répertorier les statuts d'actualisation des chèques AWS Trusted Advisor

L'`describe-trusted-advisor-check-refresh-statuses`exemple suivant répertorie les statuts d'actualisation de deux contrôles Trusted Advisor : Amazon S3 Bucket Permissions et IAM Use.

```
aws support describe-trusted-advisor-check-refresh-statuses \
    --check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

Sortie :

```
{
    "statuses": [
        {
            "checkId": "Pfx0RwqBli",
            "status": "none",
            "millisUntilNextRefreshable": 0
        },
        {
            "checkId": "zXCkfM1nI3",
            "status": "none",
            "millisUntilNextRefreshable": 0
        }
    ]
}
```

Pour plus d'informations, consultez AWS Trusted Advisor dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeTrustedAdvisorCheckRefreshStatuseslа section Référence des AWS CLI commandes.

`describe-trusted-advisor-check-result`

L'exemple de code suivant montre comment utiliser`describe-trusted-advisor-check-result`.

AWS CLI

Pour répertorier les résultats d'un check AWS Trusted Advisor

L'`describe-trusted-advisor-check-result`exemple suivant répertorie les résultats du contrôle d'utilisation IAM.

```
aws support describe-trusted-advisor-check-result \
    --check-id "zXCkfM1nI3"
```

Sortie :

```
{
    "result": {
        "checkId": "zXCkfM1nI3",
        "timestamp": "2020-05-13T21:38:05Z",
        "status": "ok",
        "resourcesSummary": {
            "resourcesProcessed": 1,
            "resourcesFlagged": 0,
            "resourcesIgnored": 0,
            "resourcesSuppressed": 0
        },
        "categorySpecificSummary": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "flaggedResources": [
            {
                "status": "ok",
                "resourceId": "47DEQpj8HBSa-_TImW-5JCeuQeRkm5NMpJWZEXAMPLE",
                "isSuppressed": false
            }
        ]
    }
}
```

Pour plus d'informations, consultez AWS Trusted Advisor dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeTrustedAdvisorCheckResultla section Référence des AWS CLI commandes.

`describe-trusted-advisor-check-summaries`

L'exemple de code suivant montre comment utiliser`describe-trusted-advisor-check-summaries`.

AWS CLI

Pour répertorier les résumés des contrôles effectués par AWS Trusted Advisor

L'`describe-trusted-advisor-check-summaries`exemple suivant répertorie les résultats de deux contrôles effectués par Trusted Advisor : Amazon S3 Bucket Permissions et IAM Use.

```
aws support describe-trusted-advisor-check-summaries \
    --check-ids "Pfx0RwqBli" "zXCkfM1nI3"
```

Sortie :

```
{
    "summaries": [
        {
            "checkId": "Pfx0RwqBli",
            "timestamp": "2020-05-13T21:38:12Z",
            "status": "ok",
            "hasFlaggedResources": true,
            "resourcesSummary": {
                "resourcesProcessed": 44,
                "resourcesFlagged": 0,
                "resourcesIgnored": 0,
                "resourcesSuppressed": 0
            },
            "categorySpecificSummary": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            }
        },
        {
            "checkId": "zXCkfM1nI3",
            "timestamp": "2020-05-13T21:38:05Z",
            "status": "ok",
            "hasFlaggedResources": true,
            "resourcesSummary": {
                "resourcesProcessed": 1,
                "resourcesFlagged": 0,
                "resourcesIgnored": 0,
                "resourcesSuppressed": 0
            },
            "categorySpecificSummary": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
```

```
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez AWS Trusted Advisor dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeTrustedAdvisorCheckSummariesla section Référence des AWS CLI commandes.

`describe-trusted-advisor-checks`

L'exemple de code suivant montre comment utiliser`describe-trusted-advisor-checks`.

AWS CLI

Pour répertorier les chèques AWS Trusted Advisor disponibles

L'`describe-trusted-advisor-checks`exemple suivant répertorie les chèques Trusted Advisor disponibles sur votre AWS compte. Ces informations incluent le nom, l'identifiant, la description, la catégorie et les métadonnées du chèque. Notez que la sortie est raccourcie pour des raisons de lisibilité.

```
aws support describe-trusted-advisor-checks \
    --language "en"
```

Sortie :

```
{
    "checks": [
        {
            "id": "zXCkfM1nI3",
            "name": "IAM Use",
            "description": "Checks for your use of AWS Identity and Access
 Management (IAM). You can use IAM to create users, groups, and roles in AWS, and
 you can use permissions to control access to AWS resources. \n<br>\n<br>\n<b>Alert
 Criteria</b><br>\nYellow: No IAM users have been created for this account.\n<br>
\n<br>\n<b>Recommended Action</b><br>\nCreate one or more IAM users and groups in
 your account. You can then create additional users whose permissions are limited
 to perform specific tasks in your AWS environment. For more information, see <a
```

```
 href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAMGettingStarted.html\"
 target=\"_blank\">Getting Started</a>. \n<br><br>\n<b>Additional Resources</b><br>
\n<a href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAM_Introduction.html\"
 target=\"_blank\">What Is IAM?</a>",
            "category": "security",
            "metadata": []
        }
    ]
}
```

Pour plus d'informations, consultez AWS Trusted Advisor dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir DescribeTrustedAdvisorChecksla section Référence des AWS CLI commandes.

`refresh-trusted-advisor-check`

L'exemple de code suivant montre comment utiliser`refresh-trusted-advisor-check`.

AWS CLI

Pour actualiser un check AWS Trusted Advisor

L'`refresh-trusted-advisor-check`exemple suivant actualise le check Amazon S3 Bucket Permissions Trusted Advisor dans votre AWS compte.

```
aws support refresh-trusted-advisor-check \
    --check-id "Pfx0RwqBli"
```

Sortie :

```
{
    "status": {
        "checkId": "Pfx0RwqBli",
        "status": "enqueued",
        "millisUntilNextRefreshable": 3599992
    }
}
```

Pour plus d'informations, consultez AWS Trusted Advisor dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir RefreshTrustedAdvisorCheckla section Référence des AWS CLI commandes.

`resolve-case`

L'exemple de code suivant montre comment utiliser`resolve-case`.

AWS CLI

Pour résoudre un dossier d'assistance

L'`resolve-case`exemple suivant résout une demande d'assistance dans votre AWS compte.

```
aws support resolve-case \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

Sortie :

```
{
    "finalCaseStatus": "resolved",
    "initialCaseStatus": "work-in-progress"
}
```

Pour plus d'informations, consultez la section Gestion des dossiers dans le Guide de l'utilisateur du AWS Support.

- Pour plus de détails sur l'API, voir ResolveCasela section Référence des AWS CLI commandes.

Exemples d'Amazon SWF utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon SWF.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`count-closed-workflow-executions`

L'exemple de code suivant montre comment utiliser`count-closed-workflow-executions`.

AWS CLI

Comptage des exécutions de flux de travail fermés

Vous pouvez l'utiliser `swf count-closed-workflow-executions` pour récupérer le nombre d'exécutions de flux de travail fermées pour un domaine donné. Vous pouvez définir des filtres pour compter des classes d'exécutions spécifiques.

Les `--start-time-filter` arguments `--domain` et l'`--close-time-filter`un ou l'autre sont obligatoires. Tous les autres arguments sont facultatifs.

```
aws swf count-closed-workflow-executions \
    --domain DataFrobtzz \
    --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
 1370044800 }"
```

Sortie :

```
{
    "count": 2,
    "truncated": false
}
```

Si « tronqué » est « tronqué »`true`, « count » représente le nombre maximum pouvant être renvoyé par Amazon SWF. Tous les autres résultats sont tronqués.

Pour réduire le nombre de résultats renvoyés, vous pouvez :

modifiez les `--start-time-filter` valeurs `--close-time-filter` ou pour réduire la plage de temps recherchée. Chacune de ces options s'exclut mutuellement : vous ne pouvez en spécifier qu'une dans une demande. Utilisez les `--type-filter` arguments `--close-`

`status-filter--execution-filter`, `--tag-filter` ou pour filtrer davantage les résultats. Cependant, ces arguments s'excluent également mutuellement.

Voir également le CountClosedWorkflowExecutionsdocument de référence sur l'API Amazon Simple Workflow Service

- Pour plus de détails sur l'API, reportez-vous CountClosedWorkflowExecutionsà la section Référence des AWS CLI commandes.

`count-open-workflow-executions`

L'exemple de code suivant montre comment utiliser`count-open-workflow-executions`.

AWS CLI

Comptage des exécutions de flux de travail ouverts

Vous pouvez l'utiliser `swf count-open-workflow-executions` pour récupérer le nombre d'exécutions de flux de travail ouvertes pour un domaine donné. Vous pouvez définir des filtres pour compter des classes d'exécutions spécifiques.

Les `--start-time-filter` arguments `--domain` et sont obligatoires. Tous les autres arguments sont facultatifs.

```
aws swf count-open-workflow-executions \
    --domain DataFrobtzz \
    --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
 1370044800 }"
```

Sortie :

```
{
    "count": 4,
    "truncated": false
}
```

Si « tronqué » est « tronqué »`true`, « count » représente le nombre maximum pouvant être renvoyé par Amazon SWF. Tous les autres résultats sont tronqués.

Pour réduire le nombre de résultats renvoyés, vous pouvez :

modifiez les `--start-time-filter` valeurs pour réduire la plage de temps recherchée. Utilisez les `--type-filter` arguments `--close-status-filter--execution-filter`, `--tag-filter` ou pour filtrer davantage les résultats. Chacune de ces options s'exclut mutuellement : vous ne pouvez en spécifier qu'une dans une demande.

Pour plus d'informations, consultez CountOpenWorkflowExecutions le manuel Amazon Simple Workflow Service API Reference

- Pour plus de détails sur l'API, reportez-vous CountOpenWorkflowExecutionsà la section Référence des AWS CLI commandes.

`deprecate-domain`

L'exemple de code suivant montre comment utiliser`deprecate-domain`.

AWS CLI

Dépréciation d'un domaine

Pour rendre un domaine obsolète (vous pouvez toujours le voir mais vous ne pouvez pas créer de nouvelles exécutions de flux de travail ou y inscrire des types), utilisez `swf deprecate-domain`. Un seul paramètre est obligatoire, `--name`, qui prend le nom du domaine à rendre obsolète.

```
aws swf deprecate-domain \
    --name MyNeatNewDomain ""
```

Tout comme avec `register-domain`, aucune sortie n'est renvoyée. Cependant, si `list-domains` vous avez l'habitude de consulter les domaines enregistrés, vous verrez que le domaine est obsolète et n'apparaît plus dans les données renvoyées.

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
                {
```

```
                    "status": "REGISTERED",
                    "name": "erontest"
                }
            ]
        }
```

Si vous utilisez `--registration-status DEPRECATED` avec`list-domains`, vous verrez votre domaine obsolète.

```
aws swf list-domains \
    --registration-status DEPRECATED
        {
            "domainInfos": [
                {
                    "status": "DEPRECATED",
                    "name": "MyNeatNewDomain"
                }
            ]
        }
```

Vous pouvez toujours l'utiliser `describe-domain` pour obtenir des informations sur un domaine obsolète.

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

Voir également le DeprecateDomaindocument de référence sur l'API Amazon Simple Workflow Service

- Pour plus de détails sur l'API, reportez-vous DeprecateDomainà la section Référence des AWS CLI commandes.

`describe-domain`

L'exemple de code suivant montre comment utiliser`describe-domain`.

AWS CLI

Obtenir des informations sur un domaine

Pour obtenir des informations détaillées sur un domaine en particulier, utilisez la `swf describe-domain` commande. Il y a un paramètre obligatoire : `--name`, qui prend le nom du domaine sur lequel vous souhaitez obtenir des informations.

```
aws swf describe-domain \
    --name DataFrobotz
        {
            "domainInfo": {
                "status": "REGISTERED",
                "name": "DataFrobotz"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "1"
            }
        }
```

Vous pouvez également l'utiliser `describe-domain` pour obtenir des informations sur les domaines obsolètes.

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

Voir également le DescribeDomaindocument de référence sur l'API Amazon Simple Workflow Service

- Pour plus de détails sur l'API, reportez-vous DescribeDomainà la section Référence des AWS CLI commandes.

`list-activity-types`

L'exemple de code suivant montre comment utiliser`list-activity-types`.

AWS CLI

Lister les types d'activités

Pour obtenir la liste des types d'activités d'un domaine, utilisez`swf list-activity-types`. Les `--registration-status` arguments `--domain` et sont obligatoires.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED
```

Sortie :

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.451,
            "activityType": {
                "version": "1",
                "name": "confirm-user-email"
            },
            "description": "subscribe confirm-user-email activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.709,
            "activityType": {
                "version": "1",
                "name": "confirm-user-phone"
            },
            "description": "subscribe confirm-user-phone activity"
        },
        {
            "status": "REGISTERED",
```

```
            "creationDate": 1371454149.871,
            "activityType": {
                "version": "1",
                "name": "get-subscription-info"
            },
            "description": "subscribe get-subscription-info activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.085,
            "activityType": {
                "version": "1",
                "name": "subscribe-user-sns"
            },
            "description": "subscribe subscribe-user-sns activity"
        }
    ]
}
```

Vous pouvez utiliser `--name` cet argument pour sélectionner uniquement les types d'activité portant un nom particulier :

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --name "send-subscription-success"
```

Sortie :

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
```

```
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

Pour récupérer les résultats sous forme de pages, vous pouvez définir l'`--maximum-page-size`argument. Si le nombre de résultats renvoyés est supérieur à ce que peut contenir une page de résultats, un nextPageToken « » sera renvoyé dans le jeu de résultats :

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2
```

Sortie :

```
{
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDl7QNY7C3OpHo9Sz06D/GuFz1OyC73umBQ1tOPJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX4OC/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarkiqpSY1ZVveBasBvlvyU
WGAaqehiDz7/JzLT/wWNNUMOd+Nhe",
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.451,
            "activityType": {
                "version": "1",
                "name": "confirm-user-email"
            },
            "description": "subscribe confirm-user-email activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.709,
            "activityType": {
                "version": "1",
                "name": "confirm-user-phone"
```

```
            },
            "description": "subscribe confirm-user-phone activity"
        }
    ]
}
```

Vous pouvez transmettre la nextPageToken valeur au prochain appel to `list-activity-types` dans l'`--next-page-token`argument, en récupérant la page de résultats suivante :

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --next-page-token "AAAAKgAAAAEAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDl7QNY7C3OpHo9Sz06D/GuFz1OyC73umBQ1tOPJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX4OC/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarkiqpSY1ZVveBasBvlvyU
WGAaqehiDz7/JzLT/wWNNUMOd+Nhe"
```

Sortie :

```
{
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCclgCuq3J+h/
m3+vOfFqtkcjLwV5cc4OjNAzTCuq/
XcylPumGwkjbajtqpZpbqOcVNfjFxGoi0LB2Olbvv0krbUISBvlpFPmSWpDSZJsxg5UxCcweteSlFn1PNSZ/
MoinBZo8OTkjMuzcsTuKOzH9wCaR8ITcALJ3SaqHU3pyIRS5hPmFA3OLIc8zaAepjlaujo6hntNSCruB4"
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454149.871,
            "activityType": {
                "version": "1",
                "name": "get-subscription-info"
            },
            "description": "subscribe get-subscription-info activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
```

```
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

S'il reste encore d'autres résultats à renvoyer, nextPageToken « » sera renvoyé avec les résultats. Lorsqu'il n'y a plus de pages de résultats à renvoyer, « nextPageToken » ne sera pas renvoyé dans le jeu de résultats.

Vous pouvez utiliser l'`--reverse-order`argument pour inverser l'ordre des résultats renvoyés. Cela affecte également les résultats de la page.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --reverse-order
```

Sortie :

```
{
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAwXcpu5ePSyQkrC
+8WMbmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCsfGkEn7mk
+yMCgVc1JxDWmB0EH46bhcmcLmYNQihMDmUWocpr7To6/R7CLu0St1gkFayxOidJXErQW0zdNfQaIWAnF/
cwioBbXlkz1fQzmDeU3M5oYGMPQIrUqkPq7pMEW0q0lK5eDN97NzFYdZZ/rlcLDWPZhUjY",
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.085,
            "activityType": {
                "version": "1",
                "name": "subscribe-user-sns"
            },
            "description": "subscribe subscribe-user-sns activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
```

```
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

Voir également le ListActivityTypesdocument de référence sur l'API Amazon Simple Workflow Service

- Pour plus de détails sur l'API, reportez-vous ListActivityTypesà la section Référence des AWS CLI commandes.

`list-domains`

L'exemple de code suivant montre comment utiliser`list-domains`.

AWS CLI

Exemple 1 : Pour répertorier vos domaines enregistrés

L'exemple de `list-domains` commande suivant répertorie les domaines `REGISTERED` SWF que vous avez enregistrés pour votre compte.

```
aws swf list-domains \
    --registration-status REGISTERED
```

Sortie :

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    },
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

Pour plus d'informations, consultez ListDomainsle manuel Amazon Simple Workflow Service API Reference

Exemple 2 : pour répertorier vos domaines obsolètes

L'exemple de `list-domains` commande suivant répertorie les domaines `DEPRECATED` SWF que vous avez enregistrés pour votre compte. Les domaines déconseillés sont des domaines qui ne peuvent pas enregistrer de nouveaux flux de travail ou de nouvelles activités, mais qui peuvent tout de même être interrogés.

```
aws swf list-domains \
    --registration-status DEPRECATED
```

Sortie :

```
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}
```

Pour plus d'informations, consultez ListDomainsle manuel Amazon Simple Workflow Service API Reference

Exemple 3 : pour répertorier la première page des domaines enregistrés

L'exemple de `list-domains` commande suivant répertorie les domaines `REGISTERED` SWF de première page que vous avez enregistrés pour votre compte à l'aide de l'`--maximum-page-size`option.

```
aws swf list-domains \
    --registration-status REGISTERED \
    --maximum-page-size 1
```

Sortie :

```
{
    "domainInfos": [
        {
```

```
            "status": "REGISTERED",
            "name": "DataFrobotz"
        }
    ],
"nextPageToken": "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL4OCXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r4OjAGaFsGOuVbrKljvla7wdU7FYH3OlkNCP8b7PBj9SBkUyGoiAg
}
```

Pour plus d'informations, consultez ListDomainsle manuel Amazon Simple Workflow Service API Reference

Exemple 4 : Pour répertorier la page unique spécifiée des domaines enregistrés

L'exemple de `list-domains` commande suivant répertorie les domaines `REGISTERED` SWF de première page que vous avez enregistrés pour votre compte à l'aide de l'`--maximum-page-size`option.

Lorsque vous effectuez à nouveau l'appel, en fournissant cette fois la valeur de `nextPageToken` dans l'`--next-page-token`argument, vous obtiendrez une autre page de résultats.

```
aws swf list-domains \
    --registration-status REGISTERED \
    --maximum-page-size 1 \
    --next-page-token "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL4OCXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r4OjAGaFsGOuVbrKljvla7wdU7FYH3OlkNCP8b7PBj9SBkUyGoiAg
```

Sortie :

```
{
    "domainInfos": [
        {
            "status": "REGISTERED",
            "name": "erontest"
        }
    ]
}
```

Lorsqu'il n'y a pas d'autres pages de résultats à extraire, `nextPageToken` n'est pas renvoyé dans les résultats.

Pour plus d'informations, consultez ListDomainsle manuel Amazon Simple Workflow Service API Reference

- Pour plus de détails sur l'API, reportez-vous ListDomainsà la section Référence des AWS CLI commandes.

`list-workflow-types`

L'exemple de code suivant montre comment utiliser`list-workflow-types`.

AWS CLI

Liste des types de flux de travail

Pour obtenir la liste des types de flux de travail pour un domaine, utilisez`swf list-workflow-types`. Les `--registration-status` arguments `--domain` et sont obligatoires. Voici un exemple simple.

```
aws swf list-workflow-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED
```

Sortie :

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454149.598,
            "description": "DataFrobtzz subscribe workflow",
            "workflowType": {
                "version": "v3",
                "name": "subscribe"
            }
        }
    ]
}
```

De même`list-activity-types`, vous pouvez utiliser l'`--name`argument pour sélectionner uniquement les types de flux de travail portant un nom particulier, et utiliser l'`--maximum-page-`

`size`argument en coordination avec les résultats `--next-page-token` de deux pages. Pour inverser l'ordre dans lequel les résultats sont renvoyés, utilisez`--reverse-order`.

Voir également le ListWorkflowTypesdocument de référence sur l'API Amazon Simple Workflow Service

- Pour plus de détails sur l'API, reportez-vous ListWorkflowTypesà la section Référence des AWS CLI commandes.

`register-domain`

L'exemple de code suivant montre comment utiliser`register-domain`.

AWS CLI

Enregistrement d'un domaine

Vous pouvez utiliser la AWS CLI pour enregistrer de nouveaux domaines. Utilisez la commande `swf register-domain`. Deux paramètres sont obligatoires : l'un prend le nom du domaine et `--workflow-execution-retention-period-in-days` l'autre prend un entier pour spécifier le nombre de jours pendant lesquels les données d'exécution du flux de travail sont conservées sur ce domaine, jusqu'à une période maximale de 90 jours (pour plus d'informations, consultez la FAQ SWF < https://aws.amazon.com/swf/ faqs/ #retain_limit >). `--name` Les données d'exécution du flux de travail ne seront pas conservées une fois le nombre de jours spécifié écoulé.

```
aws swf register-domain \
    --name MyNeatNewDomain \
    --workflow-execution-retention-period-in-days 0
    ""
```

Lorsque vous inscrivez un domaine, aucune donnée n'est renvoyée (« »), mais vous pouvez utiliser `swf list-domains` ou `swf describe-domain` pour voir le nouveau domaine.

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
```

```
            {
                "status": "REGISTERED",
                "name": "MyNeatNewDomain"
            },
            {
                "status": "REGISTERED",
                "name": "erontest"
            }
        ]
    }
```

En utilisant `swf describe-domain` :

```
aws swf describe-domain --
name MyNeatNewDomain
{
    "domainInfo": {
        "status": "REGISTERED",
        "name": "MyNeatNewDomain"
    },
    "configuration": {
        "workflowExecutionRetentionPeriodInDays": "0"

  }
}
```

Voir également le RegisterDomaindocument de référence sur l'API Amazon Simple Workflow Service

- Pour plus de détails sur l'API, reportez-vous RegisterDomainà la section Référence des AWS CLI commandes.

`register-workflow-type`

L'exemple de code suivant montre comment utiliser`register-workflow-type`.

AWS CLI

Enregistrement d'un type de flux de travail

Pour enregistrer un type de flux de travail auprès de la AWS CLI, utilisez la `swf register-workflow-type` commande.

```
aws swf register-workflow-type \
    --domain DataFrobtzz \
    --name "MySimpleWorkflow" \
    --workflow-version "v1"
```

En cas de succès, la commande ne produit aucune sortie.

En cas d'erreur (par exemple, si vous essayez d'enregistrer deux fois le même type de flux de travail ou si vous spécifiez un domaine qui n'existe pas), vous obtiendrez une réponse au format JSON.

```
{
    "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
    "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
}
```

Les`--domain`, `--name` et `--workflow-version` sont obligatoires. Vous pouvez également définir la description du flux de travail, les délais d'expiration et la politique du flux de travail pour enfants.

Pour plus d'informations, consultez RegisterWorkflowTypele manuel Amazon Simple Workflow Service API Reference

- Pour plus de détails sur l'API, reportez-vous RegisterWorkflowTypeà la section Référence des AWS CLI commandes.

Exemples de Systems Manager utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with Systems Manager.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`add-tags-to-resource`

L'exemple de code suivant montre comment utiliser`add-tags-to-resource`.

AWS CLI

Exemple 1 : ajout de balises à une fenêtre de maintenance

L’exemple de `add-tags-to-resource` suivant ajoute une balise à la fenêtre de maintenance spécifiée.

```
aws ssm add-tags-to-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "mw-03eb9db428EXAMPLE" \
    --tags "Key=Stack,Value=Production"
```

Cette commande ne produit aucun résultat.

Exemple 2 : ajout de balises à un paramètre

L’exemple de `add-tags-to-resource` suivant ajoute deux balises au paramètre spécifié.

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "My-Parameter" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

Cette commande ne produit aucun résultat.

Exemple 3 : ajout de balises à un document SSM

L’exemple de `add-tags-to-resource` suivant ajoute une balise au document spécifié.

```
aws ssm add-tags-to-resource \
    --resource-type "Document" \
    --resource-id "My-Document" \
    --tags "Key=Quarter,Value=Q322"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Balisage des ressources Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous AddTagsToResourceà la section Référence des AWS CLI commandes.

`associate-ops-item-related-item`

L'exemple de code suivant montre comment utiliser`associate-ops-item-related-item`.

AWS CLI

Pour associer un article connexe

L'`associate-ops-item-related-item`exemple suivant associe un élément associé au OpsItem.

```
aws ssm associate-ops-item-related-item \
    --ops-item-id "oi-649fExample" \
    --association-type "RelatesTo" \
    --resource-type "AWS::SSMIncidents::IncidentRecord" \
    --resource-uri "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/c2bde883-f7d5-343a-b13a-bf5fe9ea689f"
```

Sortie :

```
{
    "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"
}
```

Pour plus d'informations, voir Working with Incident Manager incidents OpsCenter dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous AssociateOpsItemRelatedItemà la section Référence des AWS CLI commandes.

`cancel-command`

L'exemple de code suivant montre comment utiliser`cancel-command`.

AWS CLI

Exemple 1 : annulation d'une commande pour toutes les instances

L'exemple de `cancel-command` suivant tente d'annuler la commande spécifiée qui est déjà en cours d'exécution pour toutes les instances.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

Cette commande ne produit aucun résultat.

Exemple 2 : annulation d'une commande pour des instances spécifiques

L'exemple de `cancel-command` suivant tente d'annuler une commande pour l'instance spécifiée uniquement.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE" \
    --instance-ids "i-02573cafcfEXAMPLE"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Balisage des paramètres Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CancelCommandà la section Référence des AWS CLI commandes.

`cancel-maintenance-window-execution`

L'exemple de code suivant montre comment utiliser`cancel-maintenance-window-execution`.

AWS CLI

Pour annuler l'exécution d'une fenêtre de maintenance

Cet `cancel-maintenance-window-execution` exemple arrête l'exécution de la fenêtre de maintenance spécifiée qui est déjà en cours.

```
aws ssm cancel-maintenance-window-execution \
```

```
    --window-execution-id j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

Sortie :

```
{
    "WindowExecutionId": "j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"
}
```

Pour plus d'informations, consultez les didacticiels Windows (AWS CLI) de maintenance de Systems Manager dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CancelMaintenanceWindowExecutionà la section Référence des AWS CLI commandes.

`create-activation`

L'exemple de code suivant montre comment utiliser`create-activation`.

AWS CLI

Pour créer une activation d’instance gérée

L’exemple de `create-activation` suivant crée une activation d’instance gérée.

```
aws ssm create-activation \
    --default-instance-name "HybridWebServers" \
    --iam-role "HybridWebServersRole" \
    --registration-limit 5
```

Sortie :

```
{
    "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
    "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

Pour plus d’informations, consultez Étape 4 : créer une activation d’instance gérée pour un environnement hybride dans le Guide de l’utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CreateActivationà la section Référence des AWS CLI commandes.

`create-association-batch`

L'exemple de code suivant montre comment utiliser`create-association-batch`.

AWS CLI

Pour créer plusieurs associations

Cet exemple associe un document de configuration à plusieurs instances. La sortie renvoie une liste des opérations réussies et échouées, le cas échéant.

Commande :

```
aws ssm create-association-batch --entries "Name=AWS-
UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-
UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

Sortie :

```
{
  "Successful": [
      {
          "Name": "AWS-UpdateSSMAgent",
          "InstanceId": "i-1234567890abcdef0",
          "AssociationVersion": "1",
          "Date": 1550504725.007,
          "LastUpdateAssociationDate": 1550504725.007,
          "Status": {
              "Date": 1550504725.007,
              "Name": "Associated",
              "Message": "Associated with AWS-UpdateSSMAgent"
          },
          "Overview": {
              "Status": "Pending",
              "DetailedStatus": "Creating"
          },
          "DocumentVersion": "$DEFAULT",
          "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
          "Targets": [
              {
                  "Key": "InstanceIds",
                  "Values": [
                      "i-1234567890abcdef0"
                  ]
```

```
                }
            ]
        },
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-9876543210abcdef0",
            "AssociationVersion": "1",
            "Date": 1550504725.057,
            "LastUpdateAssociationDate": 1550504725.057,
            "Status": {
                "Date": 1550504725.057,
                "Name": "Associated",
                "Message": "Associated with AWS-UpdateSSMAgent"
            },
            "Overview": {
                "Status": "Pending",
                "DetailedStatus": "Creating"
            },
            "DocumentVersion": "$DEFAULT",
            "AssociationId": "9c9f7f20-5154-4fed-a83e-0123456789ab",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-9876543210abcdef0"
                    ]
                }
            ]
        }
    ],
    "Failed": []
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAssociationBatchà la section Référence des AWS CLI commandes.

`create-association`

L'exemple de code suivant montre comment utiliser`create-association`.

AWS CLI

Exemple 1 : pour associer un document à l'aide d'une instance IDs

Cet exemple associe un document de configuration à une instance, en utilisant l'instance IDs.

```
aws ssm create-association \
    --instance-id "i-0cb2b964d3e14fd9f" \
    --name "AWS-UpdateSSMAgent"
```

Sortie :

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Pour plus d'informations, consultez CreateAssociationla référence de l'API AWS Systems Manager.

Exemple 2 : association d’un document à l’aide des cibles

Cet exemple associe un document de configuration à une instance à l’aide des cibles.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

Sortie :

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Pour plus d'informations, consultez CreateAssociationla référence de l'API AWS Systems Manager.

Exemple 3 : création d’une association qui ne s’exécute qu’une seule fois

Cet exemple crée une nouvelle association qui ne s’exécute qu’une seule fois à la date et à l’heure spécifiées. Les associations créées avec une date passée ou présente (au moment du traitement, la date est passée) sont exécutées immédiatement.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
    --schedule-expression "at(2020-05-14T15:55:00)"  \
    --apply-only-at-cron-interval
```

Sortie :

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Pour plus d'informations, reportez-vous CreateAssociationà la section Référence de l'API AWS Systems Manager ou Reference : Cron and rate expressions for Systems Manager dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CreateAssociationà la section Référence des AWS CLI commandes.

`create-document`

L'exemple de code suivant montre comment utiliser`create-document`.

AWS CLI

Pour créer un document

L'exemple de `create-document` suivant crée un document Systems Manager.

```
aws ssm create-document \
    --content file://exampleDocument.yml \
    --name "Example" \
    --document-type "Automation" \
    --document-format YAML
```

Sortie :

```
{
    "DocumentDescription": {
        "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583256349.452,
        "Status": "Creating",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified, Systems
 Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
```

```
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
}
```

Pour plus d'informations, consultez Création de documents Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CreateDocumentà la section Référence des AWS CLI commandes.

`create-maintenance-window`

L'exemple de code suivant montre comment utiliser`create-maintenance-window`.

AWS CLI

Exemple 1 : création d'une fenêtre de maintenance

L'exemple de `create-maintenance-window` suivant crée une nouvelle fenêtre de maintenance qui s'exécute toutes les cinq minutes pendant une durée maximale de deux heures (selon les besoins), empêche le démarrage de nouvelles tâches dans l'heure qui suit la fin de l'exécution de la fenêtre de maintenance, autorise les cibles non associées (instances que vous n'avez pas enregistrées avec la fenêtre de maintenance) et indique, par le biais de balises personnalisées, que son créateur a l'intention de l'utiliser dans le cadre d'un tutoriel.

```
aws ssm create-maintenance-window \
    --name "My-Tutorial-Maintenance-Window" \
    --schedule "rate(5 minutes)" \
    --duration 2 --cutoff 1 \
    --allow-unassociated-targets \
    --tags "Key=Purpose,Value=Tutorial"
```

Sortie :

```
{
    "WindowId": "mw-0c50858d01EXAMPLE"
}
```

Exemple 2 : pour créer une fenêtre de maintenance qui ne s'exécute qu'une seule fois

L'exemple de `create-maintenance-window` suivant crée une nouvelle fenêtre de maintenance qui ne s'exécute qu'une seule fois à la date et à l'heure spécifiées.

```
aws ssm create-maintenance-window \
    --name My-One-Time-Maintenance-Window \
    --schedule "at(2020-05-14T15:55:00)" \
    --duration 5 \
    --cutoff 2 \
    --allow-unassociated-targets \
    --tags "Key=Environment,Value=Production"
```

Sortie :

```
{
    "WindowId": "mw-01234567890abcdef"
}
```

Pour plus d'informations, consultez Fenêtres de maintenance dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CreateMaintenanceWindowà la section Référence des AWS CLI commandes.

`create-ops-item`

L'exemple de code suivant montre comment utiliser`create-ops-item`.

AWS CLI

Pour créer un OpsItems

L'`create-ops-item`exemple suivant utilise la clé /aws/resources OperationalData pour créer un OpsItem avec une ressource associée à Amazon DynamoDB.

```
aws ssm create-ops-item \
```

```
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/resources":{"Value":"[{\"arn
\": \"arn:aws:dynamodb:us-west-2:12345678:table/OpsItems
\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Sortie :

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

Pour plus d'informations, voir Creating OpsItems in the AWS Systems Manager User Guide.

- Pour plus de détails sur l'API, reportez-vous CreateOpsItemà la section Référence des AWS CLI commandes.

`create-patch-baseline`

L'exemple de code suivant montre comment utiliser`create-patch-baseline`.

AWS CLI

Exemple 1 : création d'un référentiel de correctifs avec approbation automatique

L'exemple de `create-patch-baseline` suivant crée un référentiel de correctifs pour Windows Server qui approuve les correctifs pour un environnement de production sept jours après leur publication par Microsoft.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Sortie :

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Exemple 2 : création d'un référentiel de correctifs avec une date limite d'approbation

L'exemple de `create-patch-baseline` suivant crée un référentiel de correctifs pour Windows Server qui approuve tous les correctifs pour un environnement de production qui sont publiés au plus tard le 7 juillet 2020.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Sortie :

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Exemple 3 : création d'un référentiel de correctifs avec des règles d'approbation stockées dans un fichier JSON

L'exemple de `create-patch-baseline` suivant crée un référentiel de correctifs pour Amazon Linux 2017.09 qui approuve les correctifs pour un environnement de production sept jours après leur publication, spécifie des règles d'approbation pour le référentiel de correctifs et spécifie un référentiel personnalisé pour les correctifs.

```
aws ssm create-patch-baseline \
    --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Contenu de `my-amazon-linux-approval-rules-and-repo.json` :

```
{
    "Name": "Amazon-Linux-2017.09-Production-Baseline",
    "Description": "My approval rules patch baseline for Amazon Linux 2017.09
 instances",
    "OperatingSystem": "AMAZON_LINUX",
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ],
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveAfterDays": 7,
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "Important",
                                "Critical"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
                                "Security",
                                "Bugfix"
                            ]
                        },
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "AmazonLinux2017.09"
                            ]
                        }
                    ]
                }
            }
        ]
    },
```

```
    "Sources": [
        {
            "Name": "My-AL2017.09",
            "Products": [
                "AmazonLinux2017.09"
            ],
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/mirror.list //
nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10 \nfailovermethod=priority
 \nfastestmirror_enabled=0 \ngpgcheck=1 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-
KEY-amazon-ga \nenabled=1 \nretries=3 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}
```

Exemple 4 : création d'un référentiel de correctifs qui spécifie les correctifs approuvés et rejetés

L'exemple de `create-patch-baseline` suivant spécifie explicitement les correctifs à approuver et à rejeter en tant qu'exception aux règles d'approbation par défaut.

```
aws ssm create-patch-baseline \
    --name "Amazon-Linux-2017.09-Alpha-Baseline" \
    --description "My custom approve/reject patch baseline for Amazon Linux 2017.09
 instances" \
    --operating-system "AMAZON_LINUX" \
    --approved-patches "CVE-2018-1234567,example-pkg-EE-2018*.amzn1.noarch" \
    --approved-patches-compliance-level "HIGH" \
    --approved-patches-enable-non-security \
    --tags "Key=Environment,Value=Alpha"
```

Pour plus d'informations, consultez Créer un référentiel de correctifs personnalisé dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous CreatePatchBaselineà la section Référence des AWS CLI commandes.

`create-resource-data-sync`

L'exemple de code suivant montre comment utiliser`create-resource-data-sync`.

AWS CLI

Pour créer une synchronisation des données de ressources

Cet exemple crée une synchronisation des données de ressources. Il n'y a pas de sortie si la commande réussit.

Commande :

```
aws ssm create-resource-data-sync --sync-name "ssm-resource-data-sync" --s3-destination "BucketName=ssm-bucket,Prefix=inventory,SyncFormat=JsonSerDe,Region=us-east-1"
```

- Pour plus de détails sur l'API, reportez-vous CreateResourceDataSyncà la section Référence des AWS CLI commandes.

`delete-activation`

L'exemple de code suivant montre comment utiliser`delete-activation`.

AWS CLI

Pour supprimer une activation d'instance gérée

L'exemple de `delete-activation` suivant supprime l'activation d'une instance gérée.

```
aws ssm delete-activation \
    --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration de AWS Systems Manager pour les environnements hybrides dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteActivationà la section Référence des AWS CLI commandes.

`delete-association`

L'exemple de code suivant montre comment utiliser`delete-association`.

AWS CLI

Exemple 1 : suppression d'une association en utilisant l'ID d'association

L'exemple de `delete-association` suivant supprime l'association pour l'ID d'association spécifié. Il n'y a pas de sortie si la commande réussit.

```
aws ssm delete-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modifier et créer une nouvelle version d'une association (français non garanti) dans le AWS Guide de l'utilisateur Systems Manager (français non garanti).

Exemple 2 : suppression d'une association

L'exemple de `delete-association` suivant supprime l'association entre une instance et un document. Il n'y a pas de sortie si la commande réussit.

```
aws ssm delete-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Utilisation des associations dans Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteAssociationà la section Référence des AWS CLI commandes.

`delete-document`

L'exemple de code suivant montre comment utiliser`delete-document`.

AWS CLI

Pour supprimer un document

L'exemple de `delete-document` suivant supprime un document Systems Manager.

```
aws ssm delete-document \
    --name "Example"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Création de documents Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteDocumentà la section Référence des AWS CLI commandes.

`delete-inventory`

L'exemple de code suivant montre comment utiliser`delete-inventory`.

AWS CLI

Pour supprimer un type d'inventaire personnalisé

Cet exemple supprime un schéma d'inventaire personnalisé.

Commande :

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-
option "DeleteSchema"
```

Sortie :

```
{
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",
  "TypeName": "Custom:RackInfo",
  "DeletionSummary": {
      "TotalCount": 1,
      "RemainingCount": 1,
      "SummaryItems": [
          {
              "Version": "1.0",
              "Count": 1,
              "RemainingCount": 1
          }
      ]
  }
}
```

Pour désactiver un type d'inventaire personnalisé

Cet exemple désactive un schéma d'inventaire personnalisé.

Commande :

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-
option "DisableSchema"
```

Sortie :

```
{
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",
  "TypeName": "Custom:RackInformation",
  "DeletionSummary": {
      "TotalCount": 0,
      "RemainingCount": 0,
      "SummaryItems": []
  }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteInventoryà la section Référence des AWS CLI commandes.

`delete-maintenance-window`

L'exemple de code suivant montre comment utiliser`delete-maintenance-window`.

AWS CLI

Pour supprimer une fenêtre de maintenance

Cet exemple de `delete-maintenance-window` supprime la fenêtre de maintenance spécifiée.

```
aws ssm delete-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9"
```

Sortie :

```
{
    "WindowId":"mw-1a2b3c4d5e6f7g8h9"
}
```

Pour plus d'informations, voir Supprimer une fenêtre de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteMaintenanceWindowà la section Référence des AWS CLI commandes.

`delete-parameter`

L'exemple de code suivant montre comment utiliser`delete-parameter`.

AWS CLI

Pour supprimer un paramètre

L'exemple de `delete-parameter` suivant supprime le paramètre unique spécifié.

```
aws ssm delete-parameter \
    --name "MyParameter"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Utilisation de magasins de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteParameterà la section Référence des AWS CLI commandes.

`delete-parameters`

L'exemple de code suivant montre comment utiliser`delete-parameters`.

AWS CLI

Pour supprimer une liste de paramètres

L'`delete-parameters`exemple suivant supprime les paramètres spécifiés.

```
aws ssm delete-parameters \
    --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

Sortie :

```
{
    "DeletedParameters": [
        "MyFirstParameter",
```

```
        "MySecondParameter"
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

Pour plus d'informations, consultez Utilisation de magasins de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeleteParametersà la section Référence des AWS CLI commandes.

`delete-patch-baseline`

L'exemple de code suivant montre comment utiliser`delete-patch-baseline`.

AWS CLI

Pour supprimer un référentiel de correctifs

L'exemple de `delete-patch-baseline` suivant supprime le référentiel de correctifs spécifié.

```
aws ssm delete-patch-baseline \
    --baseline-id "pb-045f10b4f382baeda"
```

Sortie :

```
{
    "BaselineId": "pb-045f10b4f382baeda"
}
```

Pour plus d'informations, consultez Mettre à jour ou supprimer un référentiel de correctifs (console) dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeletePatchBaselineà la section Référence des AWS CLI commandes.

`delete-resource-data-sync`

L'exemple de code suivant montre comment utiliser`delete-resource-data-sync`.

AWS CLI

Pour supprimer la synchronisation des données d'une ressource

Cet exemple supprime la synchronisation des données d'une ressource. Il n'y a pas de sortie si la commande réussit.

Commande :

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- Pour plus de détails sur l'API, reportez-vous DeleteResourceDataSyncà la section Référence des AWS CLI commandes.

`deregister-managed-instance`

L'exemple de code suivant montre comment utiliser`deregister-managed-instance`.

AWS CLI

Pour désenregistrer une instance gérée

L'exemple de `deregister-managed-instance` suivant désenregistre l'instance gérée spécifiée.

```
aws ssm deregister-managed-instance \
    --instance-id 'mi-08ab247cdfEXAMPLE'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Désenregistrement des nœuds gérés dans un environnement hybride et multicloud dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeregisterManagedInstanceà la section Référence des AWS CLI commandes.

`deregister-patch-baseline-for-patch-group`

L'exemple de code suivant montre comment utiliser`deregister-patch-baseline-for-patch-group`.

AWS CLI

Pour annuler l'enregistrement d'un groupe de correctifs à partir d'un référentiel de correctifs

L'exemple de `deregister-patch-baseline-for-patch-group` suivant désenregistre le groupe de correctifs spécifié du référentiel de correctifs spécifié.

```
aws ssm deregister-patch-baseline-for-patch-group \
    --patch-group "Production" \
    --baseline-id "pb-0ca44a362fEXAMPLE"
```

Sortie :

```
{
  "PatchGroup":"Production",
  "BaselineId":"pb-0ca44a362fEXAMPLE"
}
```

Pour plus d'informations, consultez Ajouter un groupe de correctifs à un référentiel de correctifs dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeregisterPatchBaselineForPatchGroupà la section Référence des AWS CLI commandes.

`deregister-target-from-maintenance-window`

L'exemple de code suivant montre comment utiliser`deregister-target-from-maintenance-window`.

AWS CLI

Pour supprimer une cible d'une fenêtre de maintenance

L'exemple de `deregister-target-from-maintenance-window` suivant supprime la cible spécifiée de la fenêtre de maintenance spécifiée.

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Sortie :

```
{
    "WindowId":"mw-ab12cd34ef56gh78",
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Pour plus d'informations, voir Mettre à jour une fenêtre de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeregisterTargetFromMaintenanceWindowà la section Référence des AWS CLI commandes.

`deregister-task-from-maintenance-window`

L'exemple de code suivant montre comment utiliser`deregister-task-from-maintenance-window`.

AWS CLI

Pour supprimer une tâche d'une fenêtre de maintenance

L'exemple de `deregister-task-from-maintenance-window` suivant supprime la tâche spécifiée de la fenêtre de maintenance spécifiée.

```
aws ssm deregister-task-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Sortie :

```
{
    "WindowTaskId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
    "WindowId":"mw-ab12cd34ef56gh78"
}
```

Pour plus d'informations, consultez les didacticiels Windows (AWS CLI) de maintenance de Systems Manager dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DeregisterTaskFromMaintenanceWindowà la section Référence des AWS CLI commandes.

`describe-activations`

L'exemple de code suivant montre comment utiliser`describe-activations`.

AWS CLI

Pour décrire les activations

L'`describe-activations`exemple suivant répertorie les informations relatives aux activations de votre AWS compte.

```
aws ssm describe-activations
```

Sortie :

```
{
    "ActivationList": [
        {
            "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
            "Description": "Example1",
            "IamRole": "HybridWebServersRole,
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1584316800.0,
            "Expired": false,
            "CreatedDate": 1581954699.792
        },
        {
            "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
            "Description": "Example2",
            "IamRole": "HybridDatabaseServersRole",
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1580515200.0,
            "Expired": true,
            "CreatedDate": 1578064132.002
        },
    ]
}
```

Pour plus d'informations, consultez Étape 4 : créer une activation d'instance gérée pour un environnement hybride dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeActivationsà la section Référence des AWS CLI commandes.

`describe-association-execution-targets`

L'exemple de code suivant montre comment utiliser`describe-association-execution-targets`.

AWS CLI

Pour obtenir des détails sur l'exécution d'une association

L'exemple de `describe-association-execution-targets` suivant décrit l'exécution de l'association spécifiée.

```
aws ssm describe-association-execution-targets \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

Sortie :

```
{
    "AssociationExecutionTargets": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "ResourceId": "i-1234567890abcdef0",
            "ResourceType": "ManagedInstance",
            "Status": "Success",
            "DetailedStatus": "Success",
            "LastExecutionDate": 1550505538.497,
            "OutputSource": {
                "OutputSourceId": "97fff367-fc5a-4299-aed8-0123456789ab",
                "OutputSourceType": "RunCommand"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Afficher l'historique des associations dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeAssociationExecutionTargetsà la section Référence des AWS CLI commandes.

`describe-association-executions`

L'exemple de code suivant montre comment utiliser`describe-association-executions`.

AWS CLI

Exemple 1 : obtention des détails de toutes les exécutions d'une association

L'exemple de `describe-association-executions` suivant décrit toutes les exécutions de l'association spécifiée.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Sortie :

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

Pour plus d'informations, consultez Afficher l'historique des associations dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 2 : obtention du détail de toutes les exécutions d'une association après une date et heure spécifiques

L'exemple de `describe-association-executions` suivant décrit toutes les exécutions d'une association après la date et l'heure spécifiées.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

Sortie :

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

Pour plus d'informations, consultez Afficher l'historique des associations dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeAssociationExecutionsà la section Référence des AWS CLI commandes.

`describe-association`

L'exemple de code suivant montre comment utiliser`describe-association`.

AWS CLI

Exemple 1 : obtention des détails d'une association

L'exemple de `describe-association` suivant décrit l'association pour l'ID d'association spécifié.

```
aws ssm describe-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Sortie :

```
{
    "AssociationDescription": {
        "Name": "AWS-GatherSoftwareInventory",
        "AssociationVersion": "1",
        "Date": 1534864780.995,
        "LastUpdateAssociationDate": 1543235759.81,
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
                "Success": 2
            }
        },
        "DocumentVersion": "$DEFAULT",
        "Parameters": {
            "applications": [
                "Enabled"
            ],
            "awsComponents": [
                "Enabled"
            ],
            "customInventory": [
                "Enabled"
            ],
```

```
            "files": [
                ""
            ],
            "instanceDetailedInformation": [
                "Enabled"
            ],
            "networkConfig": [
                "Enabled"
            ],
            "services": [
                "Enabled"
            ],
            "windowsRegistry": [
                ""
            ],
            "windowsRoles": [
                "Enabled"
            ],
            "windowsUpdates": [
                "Enabled"
            ]
        },
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "*"
                ]
            }
        ],
        "ScheduleExpression": "rate(24 hours)",
        "LastExecutionDate": 1550501886.0,
        "LastSuccessfulExecutionDate": 1550501886.0,
        "AssociationName": "Inventory-Association"
    }
}
```


Pour plus d'informations, consultez Modifier et créer une nouvelle version d'une association (français non garanti) dans le AWS Guide de l'utilisateur Systems Manager (français non garanti).

Exemple 2 : obtention des détails d'une association pour une instance et un document spécifiques

L'exemple de `describe-association` suivant décrit l'association entre une instance et un document.

```
aws ssm describe-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Sortie :

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487876122.564,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "AssociationStatusAggregatedCount": {
                "Pending": 1
            }
        },
        "AssociationId": "d8617c07-2079-4c18-9847-1234567890ab",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487876122.564,
        "Date": 1487876122.564,
        "Targets": [
            {
                "Values": [
                    "i-1234567890abcdef0"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Pour plus d'informations, consultez Modifier et créer une nouvelle version d'une association (français non garanti) dans le AWS Guide de l'utilisateur Systems Manager (français non garanti).

- Pour plus de détails sur l'API, reportez-vous DescribeAssociationà la section Référence des AWS CLI commandes.

`describe-automation-executions`

L'exemple de code suivant montre comment utiliser`describe-automation-executions`.

AWS CLI

Pour décrire une exécution d'automatisation

L'exemple de `describe-automation-executions` suivant affiche les détails d'une exécution d'automatisation.

```
aws ssm describe-automation-executions \
    --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Sortie :

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
            "DocumentName": "AWS-StartEC2Instance",
            "DocumentVersion": "1",
            "AutomationExecutionStatus": "Success",
            "ExecutionStartTime": 1583737233.748,
            "ExecutionEndTime": 1583737234.719,
            "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
            "LogFile": "",
            "Outputs": {},
            "Mode": "Auto",
            "Targets": [],
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
            "AutomationType": "Local"
        }
    ]
}
```

Pour plus d'informations, consultez Exécution d'un flux de travail d'automatisation simple dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeAutomationExecutionsà la section Référence des AWS CLI commandes.

`describe-automation-step-executions`

L'exemple de code suivant montre comment utiliser`describe-automation-step-executions`.

AWS CLI

Exemple 1 : description de toutes les étapes d'une exécution d'automatisation

L'exemple de `describe-automation-step-executions` suivant affiche les détails des étapes d'une exécution d'automatisation.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Sortie :

```
{
    "StepExecutions": [
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1583737234.134,
            "ExecutionEndTime": 1583737234.672,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
            },
            "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
            "OverriddenParameters": {}
        }
```

```
    ]
}
```

Exemple 2 : description d'une étape spécifique d'une exécution d'automatisation

L'exemple de `describe-automation-step-executions` suivant affiche les détails d'une étape spécifique d'une exécution d'automatisation.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

Pour plus d'informations, consultez Exécution d'un flux de travail d'automatisation étape par étape (ligne de commande) dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeAutomationStepExecutionsà la section Référence des AWS CLI commandes.

`describe-available-patches`

L'exemple de code suivant montre comment utiliser`describe-available-patches`.

AWS CLI

Pour obtenir les correctifs disponibles

L'exemple de `describe-available-patches` suivant permet de récupérer les détails de tous les correctifs disponibles pour Windows Server 2019 dont la sévérité du CSEM est Critique.

```
aws ssm describe-available-patches \
    --
filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"
```

Sortie :

```
{
    "Patches": [
        {
            "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
            "ReleaseDate": 1544047205.0,
            "Title": "2018-11 Update for Windows Server 2019 for x64-based Systems
 (KB4470788)",
```

```
            "Description": "Install this update to resolve issues in Windows. For a
 complete listing of the issues that are included in this update, see the associated
 Microsoft Knowledge Base article for more information. After you install this item,
 you may have to restart your computer.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4470788",
            "MsrcNumber": "",
            "Language": "All"
        },
        {
            "Id": "c96115e1-5587-4115-b851-22baa46a3f11",
            "ReleaseDate": 1549994410.0,
            "Title": "2019-02 Security Update for Adobe Flash Player for Windows
 Server 2019 for x64-based Systems (KB4487038)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues that
 are included in this update, see the associated Microsoft Knowledge Base article.
 After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4487038",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4487038",
            "MsrcNumber": "",
            "Language": "All"
        },
        ...
    ]
}
```

Pour obtenir les détails d’un correctif spécifique

L’exemple de `describe-available-patches` suivant récupère des détails sur le correctif spécifié.

```
aws ssm describe-available-patches \
    --filters "Key=PATCH_ID,Values=KB4480979"
```

Sortie :

```
{
    "Patches": [
        {
            "Id": "680861e3-fb75-432e-818e-d72e5f2be719",
            "ReleaseDate": 1546970408.0,
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues that
 are included in this update, see the associated Microsoft Knowledge Base article.
 After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4480979",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2016",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4480979",
            "MsrcNumber": "",
            "Language": "All"
        }
    ]
}
```

Pour plus d'informations, consultez Fonctionnement des opérations du gestionnaire de correctifs dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeAvailablePatchesà la section Référence des AWS CLI commandes.

`describe-document-permission`

L'exemple de code suivant montre comment utiliser`describe-document-permission`.

AWS CLI

Pour décrire les autorisations d’un document

L’exemple de `describe-document-permission` suivant affiche des détails sur les autorisations relatives à un document Systems Manager partagé publiquement.

```
aws ssm describe-document-permission \
    --name "Example" \
    --permission-type "Share"
```

Sortie :

```
{
    "AccountIds": [
        "all"
    ],
    "AccountSharingInfoList": [
        {
            "AccountId": "all",
            "SharedDocumentVersion": "$DEFAULT"
        }
    ]
}
```

Pour plus d’informations, consultez Partager un document Systems Manager dans le Guide de l’utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeDocumentPermissionà la section Référence des AWS CLI commandes.

`describe-document`

L'exemple de code suivant montre comment utiliser`describe-document`.

AWS CLI

Pour afficher les détails d’un document

L'`describe-document`exemple suivant affiche les détails d'un document Systems Manager de votre AWS compte.

```
aws ssm describe-document \
    --name "Example"
```

Sortie :

```
{
    "Document": {
        "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583257938.266,
        "Status": "Active",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified, Systems
 Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
```

```
}
```

Pour plus d'informations, consultez Création de documents Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeDocumentà la section Référence des AWS CLI commandes.

`describe-effective-instance-associations`

L'exemple de code suivant montre comment utiliser`describe-effective-instance-associations`.

AWS CLI

Pour obtenir des détails sur les associations effectives d'une instance

L'exemple de `describe-effective-instance-associations` suivant récupère les détails des associations effectives d'une instance.

Commande :

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

Sortie :

```
{
    "Associations": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "InstanceId": "i-1234567890abcdef0",
            "Content": "{\n    \"schemaVersion\": \"1.2\",\n    \"description\":
 \"Update the Amazon SSM Agent to the latest version or specified version.\",\n
   \"parameters\": {\n        \"version\": {\n            \"default\": \"\",\n
         \"description\": \"(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agent will be updated to the latest version.\",
\n            \"type\": \"String\"\n        },\n        \"allowDowngrade\": {\n
            \"default\": \"false\",\n            \"description\": \"(Optional)
 Allow the Amazon SSM Agent service to be downgraded to an earlier version. If
 set to false, the service can be upgraded to newer versions only (default). If
 set to true, specify the earlier version.\",\n            \"type\": \"String\",
\n            \"allowedValues\": [\n                \"true\",\n
```

```
 \"false\"\n            ]\n        }\n    },\n    \"runtimeConfig\": {\n
 \"aws:updateSsmAgent\": {\n            \"properties\": [\n                {\n
              \"agentName\": \"amazon-ssm-agent\",\n                \"source\":
 \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-manifest.json\",
\n                \"allowDowngrade\": \"{{ allowDowngrade }}\",\n
 \"targetVersion\": \"{{ version }}\"\n                }\n            ]\n        }\n
    }\n}\n",
            "AssociationVersion": "1"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeEffectiveInstanceAssociationsà la section Référence des AWS CLI commandes.

`describe-effective-patches-for-patch-baseline`

L'exemple de code suivant montre comment utiliser`describe-effective-patches-for-patch-baseline`.

AWS CLI

Exemple 1 : obtention de tous les correctifs définis par un référentiel de correctifs personnalisé

L'`describe-effective-patches-for-patch-baseline`exemple suivant renvoie les correctifs définis par une ligne de base de correctifs personnalisée dans le AWS compte courant. Notez que pour un référentiel personnalisé, seul l'ID est requis pour `--baseline-id`.

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "pb-08b654cf9b9681f04"
```

Sortie :

```
{
    "EffectivePatches": [
        {
            "Patch": {
                "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
                "ReleaseDate": 1544047205.0,
                "Title": "2018-11 Update for Windows Server 2019 for x64-based
 Systems (KB4470788)",
```

```
                "Description": "Install this update to resolve issues in Windows.
For a complete listing of the issues that are included in this update, see the
associated Microsoft Knowledge Base article for more information. After you install
this item, you may have to restart your computer.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Critical",
                "KbNumber": "KB4470788",
                "MsrcNumber": "",
                "Language": "All"
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
                "ApprovalDate": 1544047205.0
            }
        },
        {
            "Patch": {
                "Id": "915a6b1a-f556-4d83-8f50-b2e75a9a7e58",
                "ReleaseDate": 1549994400.0,
                "Title": "2019-02 Cumulative Update for .NET Framework 3.5 and 4.7.2
for Windows Server 2019 for x64 (KB4483452)",
                "Description": "A security issue has been identified in a Microsoft
software product that could affect your system. You can help protect your system by
installing this update from Microsoft. For a complete listing of the issues that
are included in this update, see the associated Microsoft Knowledge Base article.
After you install this update, you may have to restart your system.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4483452",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Important",
                "KbNumber": "KB4483452",
                "MsrcNumber": "",
                "Language": "All"
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
```

```
                "ApprovalDate": 1549994400.0
            }
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Exemple 2 : pour obtenir tous les correctifs définis par une ligne de base de correctifs AWS gérée

L'`describe-effective-patches-for-patch-baseline`exemple suivant renvoie les correctifs définis par une ligne de base de correctifs AWS gérée. Notez que pour une ligne de base AWS gérée, l'ARN de référence complet est requis pour `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Comment les correctifs de sécurité sont sélectionnés dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeEffectivePatchesForPatchBaselineà la section Référence des AWS CLI commandes.

`describe-instance-associations-status`

L'exemple de code suivant montre comment utiliser`describe-instance-associations-status`.

AWS CLI

Pour décrire le statut des associations d'une instance

Cet exemple montre les détails des associations d'une instance.

Commande :

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

Sortie :

```
{
  "InstanceAssociationStatusInfos": [
      {
          "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
          "Name": "AWS-GatherSoftwareInventory",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
          "InstanceId": "i-1234567890abcdef0",
          "ExecutionDate": 1550501886.0,
          "Status": "Success",
          "ExecutionSummary": "1 out of 1 plugin processed, 1 success, 0 failed, 0
 timedout, 0 skipped. ",
          "AssociationName": "Inventory-Association"
      },
      {
          "AssociationId": "5c5a31f6-6dae-46f9-944c-0123456789ab",
          "Name": "AWS-UpdateSSMAgent",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
          "InstanceId": "i-1234567890abcdef0",
          "ExecutionDate": 1550505828.548,
          "Status": "Success",
          "DetailedStatus": "Success",
          "AssociationName": "UpdateSSMAgent"
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceAssociationsStatusà la section Référence des AWS CLI commandes.

`describe-instance-information`

L'exemple de code suivant montre comment utiliser`describe-instance-information`.

AWS CLI

Exemple 1 : description des informations relatives aux instances gérées

L’exemple de `describe-instance-information` suivant récupère les détails de chacune de vos instances gérées.

```
aws ssm describe-instance-information
```

Exemple 2 : description des informations relatives à une instance gérée spécifique

L'exemple de `describe-instance-information` suivant montre les détails de l'instance gérée `i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \
    --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Exemple 3 : description des informations sur les instances gérées avec une clé de balise spécifique

L'exemple de `describe-instance-information` suivant montre les détails des instances gérées qui ont la clé de balise `DEV`.

```
aws ssm describe-instance-information \
    --filters "Key=tag-key,Values=DEV"
```

Sortie :

```
{
    "InstanceInformationList": [
        {
            "InstanceId": "i-028ea792daEXAMPLE",
            "PingStatus": "Online",
            "LastPingDateTime": 1582221233.421,
            "AgentVersion": "2.3.842.0",
            "IsLatestVersion": true,
            "PlatformType": "Linux",
            "PlatformName": "SLES",
            "PlatformVersion": "15.1",
            "ResourceType": "EC2Instance",
            "IPAddress": "192.0.2.0",
            "ComputerName": "ip-198.51.100.0.us-east-2.compute.internal",
            "AssociationStatus": "Success",
            "LastAssociationExecutionDate": 1582220806.0,
            "LastSuccessfulAssociationExecutionDate": 1582220806.0,
            "AssociationOverview": {
                "DetailedStatus": "Success",
                "InstanceAssociationStatusAggregatedCount": {
                    "Success": 2
```

```
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez Instances gérées dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceInformationà la section Référence des AWS CLI commandes.

`describe-instance-patch-states-for-patch-group`

L'exemple de code suivant montre comment utiliser`describe-instance-patch-states-for-patch-group`.

AWS CLI

Exemple 1 : obtention des états d'instance d'un groupe de correctifs

L'exemple de `describe-instance-patch-states-for-patch-group` suivant permet d'obtenir des détails sur les états récapitulatifs des correctifs par instance pour le groupe de correctifs spécifié.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Production"
```

Sortie :

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
```

```
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        },
        {
            "InstanceId": "i-0471e04240EXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-09ca3fb51fEXAMPLE",
            "SnapshotId": "05d8ffb0-1bbe-4812-ba2d-d9b7bEXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T22:06:20.340000-07:00",
            "OperationEndTime": "2021-08-04T22:07:11.220000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

Exemple 2 : obtention des états d'instance d'un groupe de correctifs avec plus de cinq correctifs manquants

L'exemple de `describe-instance-patch-states-for-patch-group` suivant permet d'obtenir des détails sur les états récapitulatifs des correctifs pour le groupe de correctifs spécifié pour les instances ayant plus de cinq correctifs manquants.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=MissingCount,Type=GreaterThan,Values=5 \
    --patch-group "Production"
```

Sortie :

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 46,
            "InstalledOtherCount": 4,
            "InstalledPendingRebootCount": 1,
            "InstalledRejectedCount": 1,
            "MissingCount": 7,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 232,
            "NotApplicableCount": 654,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 1
        }
    ]
}
```

Exemple 3 : obtention des états d'instance pour un groupe de correctifs avec moins de dix instances qui nécessitent un redémarrage

L'exemple de `describe-instance-patch-states-for-patch-group` suivant récupère les détails des états récapitulatifs des correctifs pour le groupe de correctifs spécifié pour les instances avec moins de dix instances qui nécessitent un redémarrage.

```
aws ssm describe-instance-patch-states-for-patch-group \
```

```
    --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
    --patch-group "Production"
```

Sortie :

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "PatchGroup": "Production",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 4,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 846,
            "NotApplicableCount": 212,
            "OperationStartTime": "2021-08-046T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

Pour plus d'informations, consultez Comprendre les valeurs des états de conformité des correctifs dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeInstancePatchStatesForPatchGroupà la section Référence des AWS CLI commandes.

`describe-instance-patch-states`

L'exemple de code suivant montre comment utiliser`describe-instance-patch-states`.

AWS CLI

Pour obtenir les états récapitulatifs des correctifs pour les instances

Cet exemple de `describe-instance-patch-states` permet d'obtenir les états récapitulatifs des correctifs pour une instance.

```
aws ssm describe-instance-patch-states \
    --instance-ids "i-1234567890abcdef0"
```

Sortie :

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PatchGroup": "my-patch-group",
            "BaselineId": "pb-0713accee01234567",
            "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
            "CriticalNonCompliantCount": 2,
            "SecurityNonCompliantCount": 2,
            "OtherNonCompliantCount": 1,
            "InstalledCount": 123,
            "InstalledOtherCount": 334,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 1,
            "FailedCount": 2,
            "UnreportedNotApplicableCount": 11,
            "NotApplicableCount": 2063,
            "OperationStartTime": "2021-05-03T11:00:56-07:00",
            "OperationEndTime": "2021-05-03T11:01:09-07:00",
            "Operation": "Scan",
            "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
            "RebootOption": "RebootIfNeeded"
        }
    ]
}
```

Pour plus d'informations, consultez À propos de la conformité des correctifs dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeInstancePatchStatesà la section Référence des AWS CLI commandes.

`describe-instance-patches`

L'exemple de code suivant montre comment utiliser`describe-instance-patches`.

AWS CLI

Exemple 1 : obtention des détails de l’état des correctifs d’une instance

L’exemple de `describe-instance-patches` suivant récupère les détails des correctifs pour l’instance spécifiée.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0"
```

Sortie :

```
{
    "Patches": [
        {
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "KBId": "KB4480979",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2019-01-09T00:00:00+00:00"
        },
        {
            "Title": "",
            "KBId": "KB4481031",
            "Classification": "",
            "Severity": "",
            "State": "InstalledOther",
            "InstalledTime": "2019-02-08T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Exemple 2 : obtention d'une liste de correctifs dans l'état « Manquant » pour une instance

L'exemple de `describe-instance-patches` suivant permet d'obtenir des informations sur les correctifs qui sont dans l'état Manquant pour l'instance spécifiée.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Missing
```

Sortie :

```
{
    "Patches": [
        {
            "Title": "Windows Malicious Software Removal Tool x64 - February 2019
 (KB890830)",
            "KBId": "KB890830",
            "Classification": "UpdateRollups",
            "Severity": "Unspecified",
            "State": "Missing",
            "InstalledTime": "1970-01-01T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Pour plus d'informations, consultez À propos des états de conformité des correctifs dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 3 : pour obtenir la liste des correctifs installés depuis une spécification InstalledTime pour une instance

L'exemple de `describe-instance-patches` suivant permet d'obtenir des informations sur les correctifs installés depuis une date donnée pour l'instance spécifiée en combinant l'utilisation de `--filters` et de `--query`.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Installed \
    --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

Sortie :

```
{
    "Patches": [
        {
            "Title": "2023-03 Cumulative Update for Windows Server 2019 (1809) for
 x64-based Systems (KB5023702)",
            "KBId": "KB5023702",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2023-03-16T11:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstancePatchesà la section Référence des AWS CLI commandes.

`describe-inventory-deletions`

L'exemple de code suivant montre comment utiliser`describe-inventory-deletions`.

AWS CLI

Pour obtenir des suppressions d'inventaire

Cet exemple permet de récupérer les détails des opérations de suppression d'inventaire.

Commande :

```
aws ssm describe-inventory-deletions
```

Sortie :

```
{
  "InventoryDeletions": [
      {
          "DeletionId": "6961492a-8163-44ec-aa1e-01234567850",
          "TypeName": "Custom:RackInformation",
```

```
            "DeletionStartTime": 1550254911.0,
            "LastStatus": "InProgress",
            "LastStatusMessage": "The Delete is in progress",
            "DeletionSummary": {
                "TotalCount": 0,
                "RemainingCount": 0,
                "SummaryItems": []
            },
            "LastStatusUpdateTime": 1550254911.0
        },
        {
            "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
            "TypeName": "Custom:RackInfo",
            "DeletionStartTime": 1550254859.0,
            "LastStatus": "InProgress",
            "LastStatusMessage": "The Delete is in progress",
            "DeletionSummary": {
                "TotalCount": 1,
                "RemainingCount": 1,
                "SummaryItems": [
                    {
                        "Version": "1.0",
                        "Count": 1,
                        "RemainingCount": 1
                    }
                ]
            },
            "LastStatusUpdateTime": 1550254859.0
        }
    ]
}
```

Pour obtenir les détails d'une suppression d'inventaire spécifique

Cet exemple permet de récupérer les détails d'une opération de suppression d'inventaire spécifique.

Commande :

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-
b1c6-987654321c4d"
```

Sortie :

```
{
  "InventoryDeletions": [
      {
          "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
          "TypeName": "Custom:RackInfo",
          "DeletionStartTime": 1550254859.0,
          "LastStatus": "InProgress",
          "LastStatusMessage": "The Delete is in progress",
          "DeletionSummary": {
              "TotalCount": 1,
              "RemainingCount": 1,
              "SummaryItems": [
                  {
                      "Version": "1.0",
                      "Count": 1,
                      "RemainingCount": 1
                  }
              ]
          },
          "LastStatusUpdateTime": 1550254859.0
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInventoryDeletionsà la section Référence des AWS CLI commandes.

`describe-maintenance-window-execution-task-invocations`

L'exemple de code suivant montre comment utiliser`describe-maintenance-window-execution-task-invocations`.

AWS CLI

Pour obtenir les invocations de tâches spécifiques effectuées pour l’exécution d’une tâche de fenêtre de maintenance

L’exemple de `describe-maintenance-window-execution-task-invocations` suivant répertorie les invocations de la tâche spécifiée exécutée dans le cadre de l’exécution de la fenêtre de maintenance spécifiée.

```
aws ssm describe-maintenance-window-execution-task-invocations \
```

```
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

Sortie :

```
{
    "WindowExecutionTaskInvocationIdentities": [
        {
            "Status": "SUCCESS",
            "Parameters": "{\"documentName\":\"AWS-RunShellScript\",\"instanceIds\":
[\"i-0000293ffd8c57862\"],\"parameters\":{\"commands\":[\"df\"]},\"maxConcurrency\":
\"1\",\"maxErrors\":\"1\"}",
            "InvocationId": "e274b6e1-fe56-4e32-bd2a-8073c6381d8b",
            "StartTime": 1487692834.723,
            "EndTime": 1487692834.871,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2a638355",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d384503b6586"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les tâches et les exécutions de tâches (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowExecutionTaskInvocationsà la section Référence des AWS CLI commandes.

`describe-maintenance-window-execution-tasks`

L'exemple de code suivant montre comment utiliser`describe-maintenance-window-execution-tasks`.

AWS CLI

Pour répertorier toutes les tâches associées à l'exécution d'une fenêtre de maintenance

L'exemple de `ssm describe-maintenance-window-execution-tasks` suivant répertorie les tâches associées à l'exécution de la fenêtre de maintenance spécifiée.

```
aws ssm describe-maintenance-window-execution-tasks \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Sortie :

```
{
    "WindowExecutionTaskIdentities": [
        {
            "Status": "SUCCESS",
            "TaskArn": "AWS-RunShellScript",
            "StartTime": 1487692834.684,
            "TaskType": "RUN_COMMAND",
            "EndTime": 1487692835.005,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les tâches et les exécutions de tâches (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowExecutionTasksà la section Référence des AWS CLI commandes.

`describe-maintenance-window-executions`

L'exemple de code suivant montre comment utiliser`describe-maintenance-window-executions`.

AWS CLI

Exemple 1 : répertorier toutes les exécutions pour une fenêtre de maintenance

L’exemple de `describe-maintenance-window-executions` suivant répertorie toutes les exécutions pour la fenêtre de maintenance spécifiée.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE"
```

Sortie :

```
{
    "WindowExecutions": [
        {
```

```
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
            "Status": "IN_PROGRESS",
            "StartTime": "2021-08-04T11:00:00.000000-07:00"

        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "ff75b750-4834-4377-8f61-b3cadEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": "2021-08-03T11:00:00.000000-07:00",
            "EndTime": "2021-08-03T11:37:21.450000-07:00"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "One or more tasks in the orchestration failed.",
            "StartTime": "2021-08-02T11:00:00.000000-07:00",
            "EndTime": "2021-08-02T11:22:36.190000-07:00"
        }
    ]
}
```

Exemple 2 : répertorier toutes les exécutions pour une fenêtre de maintenance avant une date spécifiée

L'exemple de `describe-maintenance-window-executions` suivant répertorie toutes les exécutions pour la fenêtre de maintenance spécifiée avant la date spécifiée.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

Sortie :

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
        "Status": "FAILED",
        "StatusDetails": "One or more tasks in the orchestration failed.",
```

```
        "StartTime": "2021-08-02T11:00:00.000000-07:00",
        "EndTime": "2021-08-02T11:22:36.190000-07:00"
    }
    ]
}
```

Exemple 3 : répertorier toutes les exécutions pour une fenêtre de maintenance après une date spécifiée

L'exemple de `describe-maintenance-window-executions` suivant répertorie toutes les exécutions pour la fenêtre de maintenance spécifiée après la date spécifiée.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

Sortie :

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
        "Status": "IN_PROGRESS",
        "StartTime": "2021-08-04T11:00:00.000000-07:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations relatives aux tâches et aux exécutions de tâches (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowExecutionsà la section Référence des AWS CLI commandes.

`describe-maintenance-window-schedule`

L'exemple de code suivant montre comment utiliser`describe-maintenance-window-schedule`.

AWS CLI

Exemple 1 : pour répertorier les prochaines exécutions pour une fenêtre de maintenance

L'`describe-maintenance-window-schedule`exemple suivant répertorie toutes les exécutions à venir pour la fenêtre de maintenance spécifiée.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-ab12cd34eEXAMPLE
```

Sortie :

```
{
    "ScheduledWindowExecutions": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2020-02-19T16:00Z"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2020-02-26T16:00Z"
        },
        ...
    ]
}
```

Exemple 2 : pour répertorier toutes les exécutions à venir pour une fenêtre de maintenance avant une date spécifiée

L'`describe-maintenance-window-schedule`exemple suivant répertorie toutes les exécutions à venir pour la fenêtre de maintenance spécifiée qui ont lieu avant la date spécifiée.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

Exemple 3 : pour répertorier toutes les exécutions à venir pour une fenêtre de maintenance après une date spécifiée

L'`describe-maintenance-window-schedule`exemple suivant répertorie toutes les exécutions à venir pour la fenêtre de maintenance spécifiée qui se produisent après la date spécifiée.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

Pour plus d'informations, consultez la section Afficher les informations relatives à la maintenance de Windows (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowScheduleà la section Référence des AWS CLI commandes.

`describe-maintenance-window-targets`

L'exemple de code suivant montre comment utiliser`describe-maintenance-window-targets`.

AWS CLI

Exemple 1 : répertorier toutes les cibles d'une fenêtre de maintenance

L'exemple de `describe-maintenance-window-targets` suivant répertorie toutes les cibles d'une fenêtre de maintenance.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Sortie :

```
{
    "Targets": [
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Single instance",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "350d44e6-28cc-44e2-951f-4b2c9EXAMPLE"
        },
```

```
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Two instances in a list",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE",
                        "i-0cb2b964d3EXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "e078a987-2866-47be-bedd-d9cf4EXAMPLE"
        }
    ]
}
```

Exemple 2 : répertorier toutes les cibles d'une fenêtre de maintenance qui correspondent à une valeur d'information spécifique sur le propriétaire

L'exemple de `describe-maintenance-window-targets` suivant répertorie toutes les cibles d'une fenêtre de maintenance correspondant à une valeur spécifique.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0ecb1226ddEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

Sortie :

```
{
    "Targets": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Environment",
                    "Values": [
                        "Prod"
                    ]
                }
```

```
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "ProdTarget1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations relatives à la maintenance de Windows (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowTargetsà la section Référence des AWS CLI commandes.

`describe-maintenance-window-tasks`

L'exemple de code suivant montre comment utiliser`describe-maintenance-window-tasks`.

AWS CLI

Exemple 1 : répertorier toutes les tâches d'une fenêtre de maintenance

L'exemple de `describe-maintenance-window-tasks` suivant répertorie toutes les tâches pour la fenêtre de maintenance spécifiée.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Sortie :

```
{
    "Tasks": [
        {
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "018b31c3-2d77-4b9e-bd48-c91edEXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Restarting EC2 Instance for maintenance",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "My-Automation-Example-Task",
            "Priority": 0,
```

```
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        },
        {
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "1943dee0-0a17-4978-9bf4-3cc2fEXAMPLE",
            "TaskArn": "AWS-DisableS3BucketPublicReadWrite",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Automation task to disable read/write access on public
 S3 buckets",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Disable-S3-Public-Read-Write-Access-Automation-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

Exemple 2 : Pour répertorier toutes les tâches d'une fenêtre de maintenance qui invoque le document de RunPowerShellScript commande AWS-

L'exemple de `describe-maintenance-window-tasks` suivant répertorie toutes les tâches de la fenêtre de maintenance spécifiée qui invoque le document de commande `AWS-RunPowerShellScript`.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Sortie :

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyTask"
        }
    ]
}
```

Exemple 3 : répertorier toutes les tâches d'une fenêtre de maintenance dont la priorité est de 3

L'exemple de `describe-maintenance-window-tasks` suivant liste toutes les tâches de la fenêtre de maintenance spécifiée qui ont une `Priority` de 3.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

Sortie :

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "ee45feff-ad65-4a6c-b478-5cab8EXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "Type": "AUTOMATION",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Automation-Task",
```

```
            "Description": "A description for my Automation task"
        }
    ]
}
```

Exemple 4 : répertorier toutes les tâches d'une fenêtre de maintenance qui ont une priorité de 1 et qui utilisent Run Command

Cet exemple de `describe-maintenance-window-tasks` répertorie toutes les tâches de la fenêtre de maintenance spécifiée qui ont une `Priority` de 1 et qui utilisent `Run Command`.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Sortie :

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les fenêtres de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowTasksà la section Référence des AWS CLI commandes.

`describe-maintenance-windows-for-target`

L'exemple de code suivant montre comment utiliser`describe-maintenance-windows-for-target`.

AWS CLI

Pour répertorier toutes les fenêtres de maintenance associées à une instance spécifique

L'`describe-maintenance-windows-for-target`exemple suivant répertorie les fenêtres de maintenance associées à des cibles ou à des tâches à l'instance spécifiée.

```
aws ssm describe-maintenance-windows-for-target \
    --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \
    --resource-type INSTANCE
```

Sortie :

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c5ed765acEXAMPLE",
            "Name": "My-First-Maintenance-Window"
        }
    ]
}
```

Pour plus d'informations, consultez la section Afficher les informations relatives à la maintenance de Windows (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous DescribeMaintenanceWindowsForTargetà la section Référence des AWS CLI commandes.

`describe-maintenance-windows`

L'exemple de code suivant montre comment utiliser`describe-maintenance-windows`.

AWS CLI

Exemple 1 : répertorier toutes les fenêtres de maintenance

L'`describe-maintenance-windows`exemple suivant répertorie toutes les fenêtres de maintenance de votre AWS compte dans la région actuelle.

```
aws ssm describe-maintenance-windows
```

Sortie :

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "Name": "MyMaintenanceWindow-1",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "Schedule": "rate(180 minutes)",
            "NextExecutionTime": "2020-02-12T23:19:20.596Z"
        },
        {
            "WindowId": "mw-03eb9db428EXAMPLE",
            "Name": "MyMaintenanceWindow-2",
            "Enabled": true,
            "Duration": 3,
            "Cutoff": 1,
            "Schedule": "rate(7 days)",
            "NextExecutionTime": "2020-02-17T23:22:00.956Z"
        },
    ]
}
```

Exemple 2 : répertorier toutes les fenêtres de maintenance activées

L’exemple de `describe-maintenance-windows` suivant répertorie toutes les fenêtres de maintenance activées.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Enabled,Values=true"
```

Exemple 3 : répertorier les fenêtres de maintenance qui correspondent à un nom spécifique

Cet exemple de `describe-maintenance-windows` répertorie toutes les fenêtres de maintenance portant le nom spécifié.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Name,Values=MyMaintenanceWindow"
```

Pour plus d'informations, consultez la section Afficher les informations relatives à la maintenance de Windows (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribeMaintenanceWindowsla section Référence des AWS CLI commandes.

`describe-ops-items`

L'exemple de code suivant montre comment utiliser`describe-ops-items`.

AWS CLI

Pour répertorier un ensemble de OpsItems

L'`describe-ops-items`exemple suivant affiche une liste de toutes les offres ouvertes OpsItems dans votre AWS compte.

```
aws ssm describe-ops-items \
    --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Sortie :

```
{
    "OpsItemSummaries": [
        {
            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-03-14T17:02:46.375000-07:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",
            "Source": "SSM",
            "Status": "Open",
            "OpsItemId": "oi-7cfc5EXAMPLE",
```

```
            "Title": "SSM Maintenance Window execution failed",
            "OperationalData": {
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-SSM-maintenance-window-
execution-failed\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        },
        {
            "CreatedBy": "arn:aws:sts::1112223233444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
            "Source": "EC2",
            "Status": "Open",
            "OpsItemId": "oi-6f966EXAMPLE",
            "Title": "EC2 instance stopped",
            "OperationalData": {
                "/aws/automations": {
                    "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-RestartEC2Instance\" } ]",
                    "Type": "SearchableString"
                },
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-stopped
\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
```

```
            "Category": "Availability",
            "Severity": "3"
        }
    ]
}
```

Pour plus d'informations, reportez-vous à la section Travailler avec OpsItems dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribeOpsItemsla section Référence des AWS CLI commandes.

`describe-parameters`

L'exemple de code suivant montre comment utiliser`describe-parameters`.

AWS CLI

Exemple 1 : répertorier tous les paramètres

L'`describe-parameters`exemple suivant répertorie tous les paramètres du AWS compte courant et de la région.

```
aws ssm describe-parameters
```

Sortie :

```
{
    "Parameters": [
        {
            "Name": "MySecureStringParameter",
            "Type": "SecureString",
            "KeyId": "alias/aws/ssm",
            "LastModifiedDate": 1582155479.205,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/Admin/
Richard-Roe-Managed",
            "Description": "This is a SecureString parameter",
            "Version": 2,
            "Tier": "Advanced",
            "Policies": [
                {
                    "PolicyText": "{\"Type\":\"Expiration\",\"Version\":\"1.0\",
\"Attributes\":{\"Timestamp\":\"2020-07-07T22:30:00Z\"}}",
```

```
                    "PolicyType": "Expiration",
                    "PolicyStatus": "Pending"
                },
                {
                    "PolicyText": "{\"Type\":\"ExpirationNotification\",\"Version\":
\"1.0\",\"Attributes\":{\"Before\":\"12\",\"Unit\":\"Hours\"}}",
                    "PolicyType": "ExpirationNotification",
                    "PolicyStatus": "Pending"
                }
            ]
        },
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Alejandro-Rosalez",
            "Description": "This is a String parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "latestAmi",
            "Type": "String",
            "LastModifiedDate": 1580862415.521,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/lambda-ssm-
role/Automation-UpdateSSM-Param",
            "Version": 3,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Exemple 2 : répertorier tous les paramètres correspondant à des métadonnées spécifiques

Cet exemple de `describe-parameters` répertorie tous les paramètres correspondant à un filtre.

aws ssm describe-parameters --filters « Clé = type, valeurs = » StringList

Sortie :

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Pour plus d'informations, consultez Recherche de paramètres Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribeParameterslа section Référence des AWS CLI commandes.

`describe-patch-baselines`

L'exemple de code suivant montre comment utiliser`describe-patch-baselines`.

AWS CLI

Exemple 1 : répertorier tous les référentiels de correctifs

L'exemple de `describe-patch-baselines` suivant permet d'obtenir les détails de tous les référentiels de correctifs de votre compte dans la région actuelle.

```
aws ssm describe-patch-baselines
```

Sortie :

```
{
    "BaselineIdentities": [
        {
            "BaselineName": "AWS-SuseDefaultPatchBaseline",
            "DefaultBaseline": true,
            "BaselineDescription": "Default Patch Baseline for Suse Provided by
 AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0123fdb36e334a3b2",
            "OperatingSystem": "SUSE"
        },
        {
            "BaselineName": "AWS-DefaultPatchBaseline",
            "DefaultBaseline": false,
            "BaselineDescription": "Default Patch Baseline Provided by AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed",
            "OperatingSystem": "WINDOWS"
        },
        ...
        {
            "BaselineName": "MyWindowsPatchBaseline",
            "DefaultBaseline": true,
            "BaselineDescription": "My patch baseline for EC2 instances for Windows
 Server",
            "BaselineId": "pb-0ad00e0dd7EXAMPLE",
            "OperatingSystem": "WINDOWS"
        }
    ]
}
```

Exemple 2 : Pour répertorier toutes les lignes de base de correctifs fournies par AWS

L'`describe-patch-baselines`exemple suivant répertorie toutes les lignes de base de correctifs fournies par AWS.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[AWS]"
```

Exemple 3 : répertorier tous les référentiels de correctifs que vous possédez

L'exemple de `describe-patch-baselines` suivant répertorie tous les référentiels de correctifs personnalisés créés dans votre compte dans la région actuelle.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[Self]"
```

Pour plus d'informations, consultez À propos des référentiels de correctifs prédéfinis et personnalisés dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribePatchBaselinesla section Référence des AWS CLI commandes.

`describe-patch-group-state`

L'exemple de code suivant montre comment utiliser`describe-patch-group-state`.

AWS CLI

Pour obtenir l'état d'un groupe de correctifs

L'exemple de `describe-patch-group-state` suivant récupère le récapitulatif de haut niveau de la conformité des correctifs pour un groupe de correctifs.

```
aws ssm describe-patch-group-state \
    --patch-group "Production"
```

Sortie :

```
{
    "Instances": 21,
    "InstancesWithCriticalNonCompliantPatches": 1,
    "InstancesWithFailedPatches": 2,
    "InstancesWithInstalledOtherPatches": 3,
    "InstancesWithInstalledPatches": 21,
    "InstancesWithInstalledPendingRebootPatches": 2,
    "InstancesWithInstalledRejectedPatches": 1,
    "InstancesWithMissingPatches": 3,
    "InstancesWithNotApplicablePatches": 4,
    "InstancesWithOtherNonCompliantPatches": 1,
    "InstancesWithSecurityNonCompliantPatches": 1,
```

```
    "InstancesWithUnreportedNotApplicablePatches": 2
}
```

Pour plus d'informations, reportez-vous aux sections À propos des groupes de correctifs < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman -patch-patchgroups.html>__ et Comprendre les valeurs d'état de conformité des correctifs dans le Guide de l'utilisateur de Systems Manager.AWS

- Pour plus de détails sur l'API, voir DescribePatchGroupStatela section Référence des AWS CLI commandes.

`describe-patch-groups`

L'exemple de code suivant montre comment utiliser`describe-patch-groups`.

AWS CLI

Pour afficher les enregistrements de groupes de correctifs

L'exemple de `describe-patch-groups` suivant répertorie les enregistrements de groupes de correctifs.

```
aws ssm describe-patch-groups
```

Sortie :

```
{
    "Mappings": [
        {
            "PatchGroup": "Production",
            "BaselineIdentity": {
                "BaselineId": "pb-0123456789abcdef0",
                "BaselineName": "ProdPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Production",
                "DefaultBaseline": false
            }
        },
        {
            "PatchGroup": "Development",
            "BaselineIdentity": {
```

```
                "BaselineId": "pb-0713accee01234567",
                "BaselineName": "DevPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Development",
                "DefaultBaseline": true
            }
        },
        ...
    ]
}
```

Pour plus d'informations, consultez les sections Créer un groupe de correctifs < https:// docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-group-tagging .html>__ et Ajouter un groupe de correctifs à une ligne de base de correctifs dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribePatchGroupsla section Référence des AWS CLI commandes.

`describe-patch-properties`

L'exemple de code suivant montre comment utiliser`describe-patch-properties`.

AWS CLI

Pour répertorier la disponibilité des correctifs Amazon Linux

L'`describe-patch-properties`exemple suivant affiche une liste des produits Amazon Linux pour lesquels des correctifs sont disponibles dans votre AWS compte.

```
aws ssm describe-patch-properties \
    --operating-system AMAZON_LINUX \
    --property PRODUCT
```

Sortie :

```
{
    "Properties": [
        {
            "Name": "AmazonLinux2012.03"
        },
```

```
        {
            "Name": "AmazonLinux2012.09"
        },
        {
            "Name": "AmazonLinux2013.03"
        },
        {
            "Name": "AmazonLinux2013.09"
        },
        {
            "Name": "AmazonLinux2014.03"
        },
        {
            "Name": "AmazonLinux2014.09"
        },
        {
            "Name": "AmazonLinux2015.03"
        },
        {
            "Name": "AmazonLinux2015.09"
        },
        {
            "Name": "AmazonLinux2016.03"
        },
        {
            "Name": "AmazonLinux2016.09"
        },
        {
            "Name": "AmazonLinux2017.03"
        },
        {
            "Name": "AmazonLinux2017.09"
        },
        {
            "Name": "AmazonLinux2018.03"
        }
    ]
}
```


Pour plus d'informations, consultez À propos des référentiels de correctifs dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribePatchPropertiesla section Référence des AWS CLI commandes.

`describe-sessions`

L'exemple de code suivant montre comment utiliser`describe-sessions`.

AWS CLI

Exemple 1 : pour répertorier toutes les sessions actives du gestionnaire de session

Cet `describe-sessions` exemple extrait la liste des dernières sessions actives créées (sessions connectées et déconnectées) au cours des 30 derniers jours et démarrées par l'utilisateur spécifié. Cette commande renvoie uniquement les résultats des connexions aux cibles initiées à l'aide du gestionnaire de session. Il ne répertorie pas les connexions établies par d'autres moyens, tels que les connexions de bureau à distance ou SSH.

```
aws ssm describe-sessions \
    --state "Active" \
    --filters "key=Owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/
Shirley-Rodriguez"
```

Sortie :

```
{
    "Sessions": [
        {
            "SessionId": "John-07a16060613c408b5",
            "Target": "i-1234567890abcdef0",
            "Status": "Connected",
            "StartDate": 1550676938.352,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-
Rodriguez",
            "OutputUrl": {}
        },
        {
            "SessionId": "John-01edf534b8b56e8eb",
            "Target": "i-9876543210abcdef0",
            "Status": "Connected",
            "StartDate": 1550676842.194,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-
Rodriguez",
            "OutputUrl": {}
        }
    ]
```

```
}
```

Exemple 2 : pour répertorier toutes les sessions du gestionnaire de session terminées

Cet `describe-sessions` exemple extrait une liste des dernières sessions interrompues au cours des 30 derniers jours pour tous les utilisateurs.

```
aws ssm describe-sessions \
    --state "History"
```

Sortie :

```
{
    "Sessions": [
        {
            "SessionId": "Mary-Major-0022b1eb2b0d9e3bd",
            "Target": "i-1234567890abcdef0",
            "Status": "Terminated",
            "StartDate": 1550520701.256,
            "EndDate": 1550521931.563,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Mary-
Major"
        },
        {
            "SessionId": "Jane-Roe-0db53f487931ed9d4",
            "Target": "i-9876543210abcdef0",
            "Status": "Terminated",
            "StartDate": 1550161369.149,
            "EndDate": 1550162580.329,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Jane-Roe"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Pour plus d'informations, consultez la section Afficher l'historique des sessions dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir DescribeSessionsla section Référence des AWS CLI commandes.

`disassociate-ops-item-related-item`

L'exemple de code suivant montre comment utiliser`disassociate-ops-item-related-item`.

AWS CLI

Pour supprimer une association d'articles connexes

L'`disassociate-ops-item-related-item`exemple suivant supprime l'association entre le OpsItem et un élément connexe.

```
aws ssm disassociate-ops-item-related-item \
    --ops-item-id "oi-f99f2EXAMPLE" \
    --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, voir Working with Incident Manager incidents OpsCenter dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir DisassociateOpsItemRelatedItemla section Référence des AWS CLI commandes.

`get-automation-execution`

L'exemple de code suivant montre comment utiliser`get-automation-execution`.

AWS CLI

Pour afficher les détails d’une exécution d’automatisation

L’exemple de `get-automation-execution` suivant affiche des informations détaillées sur une exécution d’automatisation.

```
aws ssm get-automation-execution \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Sortie :

```
{
    "AutomationExecution": {
        "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
        "DocumentName": "AWS-StartEC2Instance",
```

```
        "DocumentVersion": "1",
        "ExecutionStartTime": 1583737233.748,
        "ExecutionEndTime": 1583737234.719,
        "AutomationExecutionStatus": "Success",
        "StepExecutions": [
            {
                "StepName": "startInstances",
                "Action": "aws:changeInstanceState",
                "ExecutionStartTime": 1583737234.134,
                "ExecutionEndTime": 1583737234.672,
                "StepStatus": "Success",
                "Inputs": {
                    "DesiredState": "\"running\"",
                    "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
                },
                "Outputs": {
                    "InstanceStates": [
                        "running"
                    ]
                },
                "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
                "OverriddenParameters": {}
            }
        ],
        "StepExecutionsTruncated": false,
        "Parameters": {
            "AutomationAssumeRole": [
                ""
            ],
            "InstanceId": [
                "i-0cb99161f6EXAMPLE"
            ]
        },
        "Outputs": {},
        "Mode": "Auto",
        "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
        "Targets": [],
        "ResolvedTargets": {
            "ParameterValues": [],
            "Truncated": false
        }
    }
```

```
}
```

Pour plus d'informations, reportez-vous à la section Procédure pas à pas : patcher une AMI Linux (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetAutomationExecutionla section Référence des AWS CLI commandes.

`get-calendar-state`

L'exemple de code suivant montre comment utiliser`get-calendar-state`.

AWS CLI

Exemple 1 : pour obtenir l'état actuel d'un calendrier des modifications

Cet `get-calendar-state` exemple renvoie l'état d'un calendrier à l'heure actuelle. Comme l'exemple ne précise pas d'heure, l'état actuel du calendrier est indiqué.

```
aws ssm get-calendar-state \
    --calendar-names "MyCalendar"
```

Sortie :

```
{
    "State": "OPEN",
    "AtTime": "2020-02-19T22:28:51Z",
    "NextTransitionTime": "2020-02-24T21:15:19Z"
}
```

Exemple 2 : pour obtenir l'état d'un calendrier des modifications à une heure spécifiée

Cet `get-calendar-state` exemple renvoie l'état d'un calendrier à l'heure spécifiée.

```
aws ssm get-calendar-state \
    --calendar-names "MyCalendar" \
    --at-time "2020-07-19T21:15:19Z"
```

Sortie :

```
{
```

```
    "State": "CLOSED",
    "AtTime": "2020-07-19T21:15:19Z"
}
```

Pour plus d'informations, consultez la section Get the State of the Change Calendar dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetCalendarStatela section Référence des AWS CLI commandes.

`get-command-invocation`

L'exemple de code suivant montre comment utiliser`get-command-invocation`.

AWS CLI

Pour afficher les détails d’une invocation de commande

L’exemple de `get-command-invocation` suivant répertorie toutes les invocations de la commande spécifiée sur l’instance spécifiée.

```
aws ssm get-command-invocation \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --instance-id "i-1234567890abcdef0"
```

Sortie :

```
{
    "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
    "InstanceId": "i-1234567890abcdef0",
    "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-
d6ce8EXAMPLE",
    "DocumentName": "AWS-UpdateSSMAgent",
    "DocumentVersion": "",
    "PluginName": "aws:updateSsmAgent",
    "ResponseCode": 0,
    "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",
    "ExecutionElapsedTime": "PT0.091S",
    "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",
    "Status": "Success",
    "StatusDetails": "Success",
```

```
    "StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
 update skipped\n",
    "StandardOutputUrl": "",
    "StandardErrorContent": "",
    "StandardErrorUrl": "",
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
}
```

Pour plus d'informations, consultez Comprendre les statuts de commande dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetCommandInvocationla section Référence des AWS CLI commandes.

`get-connection-status`

L'exemple de code suivant montre comment utiliser`get-connection-status`.

AWS CLI

Pour afficher le statut de connexion d'une instance gérée

Cet exemple de `get-connection-status` renvoie le statut de connexion de l'instance gérée spécifiée.

```
aws ssm get-connection-status \
    --target i-1234567890abcdef0
```

Sortie :

```
{
    "Target": "i-1234567890abcdef0",
    "Status": "connected"
}
```

- Pour plus de détails sur l'API, voir GetConnectionStatusla section Référence des AWS CLI commandes.

`get-default-patch-baseline`

L'exemple de code suivant montre comment utiliser`get-default-patch-baseline`.

AWS CLI

Exemple 1 : affichage du référentiel de correctifs par défaut de Windows

L'exemple de `get-default-patch-baseline` suivant permet de récupérer les détails du référentiel de correctifs par défaut pour Windows Server.

```
aws ssm get-default-patch-baseline
```

Sortie :

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Exemple 2 : affichage du référentiel de correctifs par défaut pour Amazon Linux

L'exemple de `get-default-patch-baseline` suivant permet de récupérer les détails du référentiel de correctifs par défaut pour Amazon Linux.

```
aws ssm get-default-patch-baseline \
    --operating-system AMAZON_LINUX
```

Sortie :

```
{
    "BaselineId": "pb-047c6eb9c8fc12345",
    "OperatingSystem": "AMAZON_LINUX"
}
```

Pour plus d'informations, voir À propos des lignes de base de correctifs prédéfinies et personnalisées < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-baselines.html>__ et Définir une ligne de base de correctifs existante par défaut dans le guide de l'utilisateur de Systems Manager.AWS

- Pour plus de détails sur l'API, voir GetDefaultPatchBaselinela section Référence des AWS CLI commandes.

`get-deployable-patch-snapshot-for-instance`

L'exemple de code suivant montre comment utiliser`get-deployable-patch-snapshot-for-instance`.

AWS CLI

Pour récupérer l'instantané actuel pour le référentiel de correctifs qu'une instance utilise

L'exemple de `get-deployable-patch-snapshot-for-instance` suivant récupère les détails de l'instantané en cours pour le référentiel de correctifs spécifié utilisé par une instance. Cette commande doit être exécutée à partir de l'instance en utilisant les informations d'identification de l'instance. Pour vous assurer qu'il utilise les informations d'identification de l'instance, exécutez `aws configure` et spécifiez uniquement la région de votre instance. Laissez les champs `Access Key` et `Secret Key` vides.

Conseil : utilisez `uuidgen` pour générer un `snapshot-id`.

```
aws ssm get-deployable-patch-snapshot-for-instance \
    --instance-id "i-1234567890abcdef0" \
    --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Sortie :

```
{
    "InstanceId": "i-1234567890abcdef0",
    "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
    "Product": "AmazonLinux2018.03",
    "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-
east-1.s3.amazonaws.com/
ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/
AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20190215T164031Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-
Amz-Credential=AKIAJ5C56P35AEBRX2QQ%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-
Amz-Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
}
```

Pour plus d'informations, consultez Nom du paramètre : ID de l'instantané dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetDeployablePatchSnapshotForInstancela section Référence des AWS CLI commandes.

`get-document`

L'exemple de code suivant montre comment utiliser`get-document`.

AWS CLI

Pour obtenir le contenu d’un document

L’exemple de `get-document` suivant affiche le contenu d’un document Systems Manager.

```
aws ssm get-document \
    --name "AWS-RunShellScript"
```

Sortie :

```
{
    "Name": "AWS-RunShellScript",
    "DocumentVersion": "1",
    "Status": "Active",
    "Content": "{\n    \"schemaVersion\":\"1.2\",\n    \"description\":\"Run a
 shell script or specify the commands to run.\",\n    \"parameters\":{\n
 \"commands\":{\n            \"type\":\"StringList\",\n            \"description
\":\"(Required) Specify a shell script or a command to run.\",\n
 \"minItems\":1,\n            \"displayType\":\"textarea\"\n        },\n
 \"workingDirectory\":{\n            \"type\":\"String\",\n            \"default
\":\"\",\n            \"description\":\"(Optional) The path to the working
 directory on your instance.\",\n            \"maxChars\":4096\n        },\n
   \"executionTimeout\":{\n            \"type\":\"String\",\n            \"default
\":\"3600\",\n            \"description\":\"(Optional) The time in seconds for a
 command to complete before it is considered to have failed. Default is 3600 (1
 hour). Maximum is 172800 (48 hours).\",\n            \"allowedPattern\":\"([1-9]
[0-9]{0,4})|(1[0-6][0-9]{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n
        }\n    },\n    \"runtimeConfig\":{\n        \"aws:runShellScript\":{\n
            \"properties\":[\n                {\n                    \"id\":
\"0.aws:runShellScript\",\n                    \"runCommand\":\"{{ commands }}\",\n
                    \"workingDirectory\":\"{{ workingDirectory }}\",\n
       \"timeoutSeconds\":\"{{ executionTimeout }}\"\n                }\n
  ]\n        }\n    }\n}\n",
    "DocumentType": "Command",
    "DocumentFormat": "JSON"
}
```

Pour plus d'informations, consultez Documents AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetDocumentla section Référence des AWS CLI commandes.

`get-inventory-schema`

L'exemple de code suivant montre comment utiliser`get-inventory-schema`.

AWS CLI

Pour afficher votre schéma d'inventaire

Cet exemple renvoie une liste de noms de types d'inventaire pour le compte.

Commande :

```
aws ssm get-inventory-schema
```

Sortie :

```
{
  "Schemas": [
      {
          "TypeName": "AWS:AWSComponent",
          "Version": "1.0",
          "Attributes": [
              {
                  "Name": "Name",
                  "DataType": "STRING"
              },
              {
                  "Name": "ApplicationType",
                  "DataType": "STRING"
              },
              {
                  "Name": "Publisher",
                  "DataType": "STRING"
              },
              {
                  "Name": "Version",
```

```
                "DataType": "STRING"
            },
            {
                "Name": "InstalledTime",
                "DataType": "STRING"
            },
            {
                "Name": "Architecture",
                "DataType": "STRING"
            },
            {
                "Name": "URL",
                "DataType": "STRING"
            }
        ]
    },
    ...
  ],
  "NextToken": "--token string truncated--"
}
```

Pour afficher le schéma d'inventaire d'un type d'inventaire spécifique

Cet exemple renvoie le schéma d'inventaire pour un type d'inventaire des AWS composants AWS :

Commande :

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- Pour plus de détails sur l'API, voir GetInventorySchemala section Référence des AWS CLI commandes.

`get-inventory`

L'exemple de code suivant montre comment utiliser`get-inventory`.

AWS CLI

Pour afficher votre inventaire

Cet exemple permet d'obtenir les métadonnées personnalisées de votre inventaire.

Commande :

```
aws ssm get-inventory
```

Sortie :

```
{
  "Entities": [
      {
          "Data": {
              "AWS:InstanceInformation": {
                  "Content": [
                      {
                          "ComputerName": "ip-172-31-44-222.us-
west-2.compute.internal",
                          "InstanceId": "i-0cb2b964d3e14fd9f",
                          "IpAddress": "172.31.44.222",
                          "AgentType": "amazon-ssm-agent",
                          "ResourceType": "EC2Instance",
                          "AgentVersion": "2.0.672.0",
                          "PlatformVersion": "2016.09",
                          "PlatformName": "Amazon Linux AMI",
                          "PlatformType": "Linux"
                      }
                  ],
                  "TypeName": "AWS:InstanceInformation",
                  "SchemaVersion": "1.0",
                  "CaptureTime": "2017-02-20T18:03:58Z"
              }
          },
          "Id": "i-0cb2b964d3e14fd9f"
      }
  ]
}
```

- Pour plus de détails sur l'API, voir GetInventorylla section Référence des AWS CLI commandes.

`get-maintenance-window-execution-task-invocation`

L'exemple de code suivant montre comment utiliser`get-maintenance-window-execution-task-invocation`.

AWS CLI

Pour obtenir des informations sur l'invocation d'une tâche pendant la fenêtre de maintenance

L'`get-maintenance-window-execution-task-invocation`exemple suivant répertorie les informations relatives à l'appel de tâches spécifié qui fait partie de l'exécution de la fenêtre de maintenance spécifiée.

```
aws ssm get-maintenance-window-execution-task-invocation \
    --window-execution-id "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE" \
    --task-id "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE" \
    --invocation-id "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE"
```

Sortie :

```
{
    "Status": "SUCCESS",
    "Parameters": "{\"comment\":\"\",\"documentName\":\"AWS-RunPowerShellScript\",
\"instanceIds\":[\"i-1234567890EXAMPLE\"],\"maxConcurrency\":\"1\",\"maxErrors\":
\"1\",\"parameters\":{\"executionTimeout\":[\"3600\"],\"workingDirectory\":[\"\"],
\"commands\":[\"echo Hello\"]},\"timeoutSeconds\":600}",
    "ExecutionId": "03b6baa0-5460-4e15-83f2-ea685EXAMPLE",
    "InvocationId": "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE",
    "StartTime": 1549998326.421,
    "TaskType": "RUN_COMMAND",
    "EndTime": 1550001931.784,
    "WindowExecutionId": "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE",
    "StatusDetails": "Failed",
    "TaskExecutionId": "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE"
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les tâches et les exécutions de tâches (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetMaintenanceWindowExecutionTaskInvocationla section Référence des AWS CLI commandes.

`get-maintenance-window-execution-task`

L'exemple de code suivant montre comment utiliser`get-maintenance-window-execution-task`.

AWS CLI

Pour obtenir des informations sur l'exécution d'une tâche de la fenêtre de maintenance

L'exemple de `get-maintenance-window-execution-task` suivant répertorie les informations relatives à une tâche qui fait partie de l'exécution de la fenêtre de maintenance spécifiée.

```
aws ssm get-maintenance-window-execution-task \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

Sortie :

```
{
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
    "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE",
    "TaskArn": "AWS-RunPatchBaseline",
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "Type": "RUN_COMMAND",
    "TaskParameters": [
        {
            "BaselineOverride": {
                "Values": [
                    ""
                ]
            },
            "InstallOverrideList": {
                "Values": [
                    ""
                ]
            },
            "Operation": {
                "Values": [
                    "Scan"
                ]
            },
            "RebootOption": {
                "Values": [
                    "RebootIfNeeded"
                ]
            },
```

```
            "SnapshotId": {
                "Values": [
                    "{{ aws:ORCHESTRATION_ID }}"
                ]
            },
            "aws:InstanceId": {
                "Values": [
                    "i-02573cafcfEXAMPLE",
                    "i-0471e04240EXAMPLE",
                    "i-07782c72faEXAMPLE"
                ]
            }
        }
    ],
    "Priority": 1,
    "MaxConcurrency": "1",
    "MaxErrors": "3",
    "Status": "SUCCESS",
    "StartTime": "2021-08-04T11:45:35.088000-07:00",
    "EndTime": "2021-08-04T11:53:09.079000-07:00"
}
```

Pour plus d'informations, consultez la section Afficher les informations relatives aux tâches et aux exécutions de tâches (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetMaintenanceWindowExecutionTaskla section Référence des AWS CLI commandes.

`get-maintenance-window-execution`

L'exemple de code suivant montre comment utiliser`get-maintenance-window-execution`.

AWS CLI

Pour obtenir des informations sur l’exécution d’une tâche de la fenêtre de maintenance

L’exemple de `get-maintenance-window-execution` suivant répertorie les informations relatives à une tâche exécutée dans le cadre de l’exécution de la fenêtre de maintenance spécifiée.

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Sortie :

```
{
    "Status": "SUCCESS",
    "TaskIds": [
        "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
    ],
    "StartTime": 1487692834.595,
    "EndTime": 1487692835.051,
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les tâches et les exécutions de tâches (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetMaintenanceWindowExecutionla section Référence des AWS CLI commandes.

`get-maintenance-window-task`

L'exemple de code suivant montre comment utiliser`get-maintenance-window-task`.

AWS CLI

Pour obtenir des informations sur une tâche de fenêtre de maintenance

L'`get-maintenance-window-task`exemple suivant permet de récupérer des informations sur la tâche de la fenêtre de maintenance spécifiée.

```
aws ssm get-maintenance-window-task \
    --window-id mw-0c5ed765acEXAMPLE \
    --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

Sortie :

```
{
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "MaxErrors": "1",
    "TaskArn": "AWS-RunPowerShellScript",
    "MaxConcurrency": "1",
```

```
    "WindowTaskId": "0e842a8d-2d44-4886-bb62-af8dcEXAMPLE",
    "TaskParameters": {},
    "Priority": 1,
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "",
            "TimeoutSeconds": 600,
            "Parameters": {
                "commands": [
                    "echo Hello"
                ],
                "executionTimeout": [
                    "3600"
                ],
                "workingDirectory": [
                    ""
                ]
            }
        }
    },
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "TaskType": "RUN_COMMAND",
    "Targets": [
        {
            "Values": [
                "84c818da-b619-4d3d-9651-946f3EXAMPLE"
            ],
            "Key": "WindowTargetIds"
        }
    ],
    "Name": "ExampleTask"
}
```

Pour plus d'informations, consultez la section Afficher les informations relatives à la maintenance de Windows (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetMaintenanceWindowTaskla section Référence des AWS CLI commandes.

`get-maintenance-window`

L'exemple de code suivant montre comment utiliser`get-maintenance-window`.

AWS CLI

Pour obtenir des informations sur une fenêtre de maintenance

L'exemple de `get-maintenance-window` suivant permet d'obtenir des informations sur la fenêtre de maintenance spécifiée.

```
aws ssm get-maintenance-window \
    --window-id "mw-03eb9db428EXAMPLE"
```

Sortie :

```
{
    "AllowUnassociatedTargets": true,
    "CreatedDate": 1515006912.957,
    "Cutoff": 1,
    "Duration": 6,
    "Enabled": true,
    "ModifiedDate": 2020-01-01T10:04:04.099Z,
    "Name": "My-Maintenance-Window",
    "Schedule": "rate(3 days)",
    "WindowId": "mw-03eb9db428EXAMPLE",
    "NextExecutionTime": "2020-02-25T00:08:15.099Z"
}
```

Pour plus d'informations, consultez la section Afficher les informations sur les fenêtres de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetMaintenanceWindowla section Référence des AWS CLI commandes.

`get-ops-item`

L'exemple de code suivant montre comment utiliser`get-ops-item`.

AWS CLI

Pour consulter les informations relatives à un OpsItem

L'`get-ops-item`exemple suivant affiche les détails relatifs à ce qui est spécifié OpsItem.

```
aws ssm get-ops-item \
```

```
    --ops-item-id oi-0b725EXAMPLE
```

Sortie :

```
{
    "OpsItem": {
        "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
        "CreatedTime": "2019-12-04T15:52:16.793000-08:00",
        "Description": "CloudWatch Event Rule SSMOpsItems-EC2-instance-terminated
 was triggered. Your EC2 instance has terminated. See below for more details.",
        "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
        "LastModifiedTime": "2019-12-04T15:52:16.793000-08:00",
        "Notifications": [],
        "RelatedOpsItems": [],
        "Status": "Open",
        "OpsItemId": "oi-0b725EXAMPLE",
        "Title": "EC2 instance terminated",
        "Source": "EC2",
        "OperationalData": {
            "/aws/automations": {
                "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-CreateManagedWindowsInstance\" }, { \"automationType\":
 \"AWS:SSM:Automation\", \"automationId\": \"AWS-CreateManagedLinuxInstance\" } ]",
                "Type": "SearchableString"
            },
            "/aws/dedup": {
                "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-terminated
\"}",
                "Type": "SearchableString"
            },
            "/aws/resources": {
                "Value": "[{\"arn\":\"arn:aws:ec2:us-east-2:111222333444:instance/
i-05adec7e97EXAMPLE\"}]",
                "Type": "SearchableString"
            },
            "event-time": {
                "Value": "2019-12-04T23:52:16Z",
                "Type": "String"
            },
            "instance-state": {
                "Value": "terminated",
```

```
                "Type": "String"
            }
        },
        "Category": "Availability",
        "Severity": "4"
    }
}
```

Pour plus d'informations, reportez-vous à la section Travailler avec OpsItems dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetOpsItemla section Référence des AWS CLI commandes.

`get-ops-summary`

L'exemple de code suivant montre comment utiliser`get-ops-summary`.

AWS CLI

Pour afficher un résumé de tous OpsItems

L'`get-ops-summary`exemple suivant affiche un résumé de tous les éléments OpsItems de votre AWS compte.

```
aws ssm get-ops-summary
```

Sortie :

```
{
    "Entities": [
        {
            "Id": "oi-4309fEXAMPLE",
            "Data": {
                "AWS:OpsItem": {
                    "CaptureTime": "2020-02-26T18:58:32.918Z",
                    "Content": [
                        {
                            "AccountId": "111222333444",
                            "Category": "Availability",
                            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "CreatedTime": "2020-02-26T19:10:44.149Z",
```

```
                            "Description": "CloudWatch Event Rule SSMOpsItems-EC2-
instance-terminated was triggered. Your EC2 instance has terminated. See below for
 more details.",
                            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "LastModifiedTime": "2020-02-26T19:10:44.149Z",
                            "Notifications": "",
                            "OperationalData": "{\"/aws/automations\":
{\"type\":\"SearchableString\",\"value\":\"[ { \\\"automationType\\\": \\
\"AWS:SSM:Automation\\\", \\\"automationId\\\": \\\"AWS-CreateManagedWindowsInstance
\\\" }, { \\\"automationType\\\": \\\"AWS:SSM:Automation\\\", \\\"automationId
\\\": \\\"AWS-CreateManagedLinuxInstance\\\" } ]\"},\"/aws/resources\":
{\"type\":\"SearchableString\",\"value\":\"[{\\\"arn\\\":\\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0acbd0800fEXAMPLE\\\"}]\"},\"/aws/dedup\":{\"type\":
\"SearchableString\",\"value\":\"{\\\"dedupString\\\":\\\"SSMOpsItems-EC2-instance-
terminated\\\"}\"}}",
                            "OpsItemId": "oi-4309fEXAMPLE",
                            "RelatedItems": "",
                            "Severity": "3",
                            "Source": "EC2",
                            "Status": "Open",
                            "Title": "EC2 instance terminated"
                        }
                    ]
                }
            }
        },
        {
            "Id": "oi-bb2a0e6a4541",
            "Data": {
                "AWS:OpsItem": {
                    "CaptureTime": "2019-11-26T19:20:06.161Z",
                    "Content": [
                        {
                            "AccountId": "111222333444",
                            "Category": "Availability",
                            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "CreatedTime": "2019-11-26T20:00:07.237Z",
                            "Description": "CloudWatch Event Rule SSMOpsItems-SSM-
maintenance-window-execution-failed was triggered. Your SSM Maintenance Window
 execution has failed. See below for more details.",
                            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
```

```
                            "LastModifiedTime": "2019-11-26T20:00:07.237Z",
                            "Notifications": "",
                            "OperationalData": "{\"/aws/resources\":{\"type
\":\"SearchableString\",\"value\":\"[{\\\"arn\\\":\\\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-0e83ba440dEXAMPLE\\\"}]\"},\"/aws/dedup\":
{\"type\":\"SearchableString\",\"value\":\"{\\\"dedupString\\\":\\\"SSMOpsItems-SSM-
maintenance-window-execution-failed\\\"}\"}}",
                            "OpsItemId": "oi-bb2a0EXAMPLE",
                            "RelatedItems": "",
                            "Severity": "3",
                            "Source": "SSM",
                            "Status": "Open",
                            "Title": "SSM Maintenance Window execution failed"
                        }
                    ]
                }
            }
        ]
}
```

Pour plus d'informations, reportez-vous à la section Travailler avec OpsItems dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetOpsSummaryla section Référence des AWS CLI commandes.

`get-parameter-history`

L'exemple de code suivant montre comment utiliser`get-parameter-history`.

AWS CLI

Pour obtenir un historique des valeurs d’un paramètre

L’exemple de `get-parameter-history` suivant répertorie l’historique des modifications pour le paramètre spécifié, y compris sa valeur.

```
aws ssm get-parameter-history \
    --name "MyStringParameter"
```

Sortie :

```
{
    "Parameters": [
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the first version of my String parameter",
            "Value": "Veni",
            "Version": 1,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156093.471,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the second version of my String parameter",
            "Value": "Vidi",
            "Version": 2,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156117.545,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the third version of my String parameter",
            "Value": "Vici",
            "Version": 3,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```


Pour plus d'informations, consultez Utilisation des versions de paramètre dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetParameterHistoryla section Référence des AWS CLI commandes.

`get-parameter`

L'exemple de code suivant montre comment utiliser`get-parameter`.

AWS CLI

Exemple 1 : affichage de la valeur d’un paramètre

L’exemple de `get-parameter` suivant répertorie la valeur de l’unique paramètre spécifié.

```
aws ssm get-parameter \
    --name "MyStringParameter"
```

Sortie :

```
{
    "Parameter": {
        "Name": "MyStringParameter",
        "Type": "String",
        "Value": "Veni",
        "Version": 1,
        "LastModifiedDate": 1530018761.888,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
}
```

Pour plus d’informations, consultez Utilisation de magasins de paramètres dans le Guide de l’utilisateur AWS Systems Manager.

Exemple 2 : pour déchiffrer la valeur d'un paramètre SecureString

L’exemple de `get-parameter` suivant déchiffre la valeur du paramètre `SecureString` spécifié.

```
aws ssm get-parameter \
    --name "MySecureStringParameter" \
    --with-decryption
```

Sortie :

```
{
    "Parameter": {
        "Name": "MySecureStringParameter",
        "Type": "SecureString",
        "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
        "Version": 2,
        "LastModifiedDate": 1582155479.205,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
        "DataType": "text"
    }
}
```

Pour plus d'informations, consultez Utilisation de magasins de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 3 : affichage de la valeur d'un paramètre à l'aide d'étiquettes

L'exemple `get-parameter` suivant répertorie la valeur du paramètre unique spécifié à l'aide d'une étiquette.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

Sortie :

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":label",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

Pour plus d'informations, consultez Utilisation des étiquettes de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 4 : affichage de la valeur d'un paramètre en utilisant les versions

L'exemple de `get-parameter` suivant répertorie la valeur de la version du paramètre unique spécifié.

```
aws ssm get-parameter \
    --name "MyParameter:2"
```

Sortie :

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":2",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

Pour plus d'informations, consultez Utilisation des étiquettes de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetParameterla section Référence des AWS CLI commandes.

`get-parameters-by-path`

L'exemple de code suivant montre comment utiliser`get-parameters-by-path`.

AWS CLI

Pour répertorier les paramètres dans un chemin spécifique

L'`get-parameters-by-path`exemple suivant répertorie les paramètres au sein de la hiérarchie spécifiée.

```
aws ssm get-parameters-by-path \
```

```
    --path "/site/newyork/department/"
```

Sortie :

```
{
    "Parameters": [
        {
            "Name": "/site/newyork/department/marketing",
            "Type": "String",
            "Value": "Floor 2",
            "Version": 1,
            "LastModifiedDate": 1530018761.888,
            "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/marketing"
        },
        {
            "Name": "/site/newyork/department/infotech",
            "Type": "String",
            "Value": "Floor 3",
            "Version": 1,
            "LastModifiedDate": 1530018823.429,
            "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/infotech"
        },
        ...
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des hiérarchies de paramètres dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetParametersByPathla section Référence des AWS CLI commandes.

`get-parameters`

L'exemple de code suivant montre comment utiliser`get-parameters`.

AWS CLI

Exemple 1 : répertorier les valeurs d’un paramètre

L’exemple de `get-parameters` suivant répertorie les valeurs des trois paramètres spécifiés.

```
aws ssm get-parameters \
    --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

Sortie :

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "Value": "alpha,beta,gamma",
            "Version": 1,
            "LastModifiedDate": 1582154764.222,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
            "DataType": "text"
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "Value": "Vici",
            "Version": 3,
            "LastModifiedDate": 1582156117.545,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
            "DataType": "text"
        }
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

Pour plus d'informations, consultez Utilisation de magasins de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 2 : répertorier les noms et les valeurs de plusieurs paramètres en utilisant l'option ``--query``

L'exemple de `get-parameters` suivant répertorie les noms et les valeurs des paramètres spécifiés.

```
aws ssm get-parameters \
```

```
    --names MyStringParameter MyStringListParameter \
    --query "Parameters[*].{Name:Name,Value:Value}"
```

Sortie :

```
[
    {
        "Name": "MyStringListParameter",
        "Value": "alpha,beta,gamma"
    },
    {
        "Name": "MyStringParameter",
        "Value": "Vidi"
    }
]
```

Pour plus d'informations, consultez Utilisation de magasins de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 3 : affichage de la valeur d'un paramètre à l'aide d'étiquettes

L'exemple `get-parameter` suivant répertorie la valeur du paramètre unique spécifié à l'aide d'une étiquette.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

Sortie :

```
{
    "Parameters": [
        {
            "Name": "MyLabelParameter",
            "Type": "String",
            "Value": "parameter by label",
            "Version": 1,
            "Selector": ":label",
            "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
            "DataType": "text"
        },
```

```
        {
            "Name": "MyVersionParameter",
            "Type": "String",
            "Value": "parameter by version",
            "Version": 2,
            "Selector": ":2",
            "LastModifiedDate": "2021-03-24T16:20:28.236000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/unlabel-param",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Pour plus d'informations, consultez Utilisation des étiquettes de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetParameterslа section Référence des AWS CLI commandes.

`get-patch-baseline-for-patch-group`

L'exemple de code suivant montre comment utiliser`get-patch-baseline-for-patch-group`.

AWS CLI

Pour afficher le référentiel de correctifs d'un groupe de correctifs

L'exemple de `get-patch-baseline-for-patch-group` suivant permet de récupérer les détails du référentiel de correctifs pour le groupe de correctifs spécifié.

```
aws ssm get-patch-baseline-for-patch-group \
    --patch-group "DEV"
```

Sortie :

```
{
    "PatchGroup": "DEV",
    "BaselineId": "pb-0123456789abcdef0",
    "OperatingSystem": "WINDOWS"
}
```

Pour plus d'informations, consultez les sections Créer un groupe de correctifs < https:// docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-group-tagging .html>__ et Ajouter un groupe de correctifs à une ligne de base de correctifs dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetPatchBaselineForPatchGroupla section Référence des AWS CLI commandes.

`get-patch-baseline`

L'exemple de code suivant montre comment utiliser`get-patch-baseline`.

AWS CLI

Pour afficher un référentiel de correctifs

L'exemple de `get-patch-baseline` suivant récupère les détails du référentiel de correctifs spécifié.

```
aws ssm get-patch-baseline \
    --baseline-id "pb-0123456789abcdef0"
```

Sortie :

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
```

```
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "PatchGroups": [
        "QA",
        "DEV"
    ],
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

Pour plus d'informations, consultez À propos des référentiels de correctifs dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetPatchBaselinela section Référence des AWS CLI commandes.

`get-service-setting`

L'exemple de code suivant montre comment utiliser`get-service-setting`.

AWS CLI

Pour récupérer le paramètre de service pour le débit du Parameter Store

`get-service-setting`Cet exemple extrait le paramètre de service actuel pour le débit du Parameter Store dans la région spécifiée.

```
aws ssm get-service-setting \
```

```
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled
```

Sortie :

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/parameter-store/high-throughput-enabled",
        "SettingValue": "false",
        "LastModifiedDate": 1555532818.578,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled",
        "Status": "Default"
    }
}
```

Pour plus d'informations, consultez la section Increasing Parameter Store Throughput dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir GetServiceSettingla section Référence des AWS CLI commandes.

`label-parameter-version`

L'exemple de code suivant montre comment utiliser`label-parameter-version`.

AWS CLI

Exemple 1 : pour ajouter une étiquette à la dernière version d'un paramètre

L'`label-parameter-version`exemple suivant ajoute une étiquette à la dernière version du paramètre spécifié.

```
aws ssm label-parameter-version \
    --name "MyStringParameter" \
    --labels "ProductionReady"
```

Sortie :

```
{
    "InvalidLabels": [],
```

```
    "ParameterVersion": 3
}
```

Pour plus d'informations, consultez Utilisation des étiquettes de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 2 : pour ajouter une étiquette à une version spécifique d'un paramètre

L'`label-parameter-version`exemple suivant ajoute une étiquette à la version spécifiée d'un paramètre.

```
aws ssm label-parameter-version \
    --name "MyStringParameter" \
    --labels "ProductionReady" \
    --parameter-version "2" --labels "DevelopmentReady"
```

Pour plus d'informations, consultez Utilisation des étiquettes de paramètres dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir LabelParameterVersionla section Référence des AWS CLI commandes.

`list-association-versions`

L'exemple de code suivant montre comment utiliser`list-association-versions`.

AWS CLI

Pour répertorier toutes les versions d'une association pour un ID d'association spécifique

L'exemple de `list-association-versions` suivant répertorie toutes les versions des associations spécifiées.

```
aws ssm list-association-versions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Sortie :

```
{
"AssociationVersions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
```

```
            "AssociationVersion": "1",
            "CreatedDate": 1550505536.726,
            "Name": "AWS-UpdateSSMAgent",
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ],
                "version": [
                    ""
                ]
            },
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

Pour plus d'informations, consultez Utilisation des associations dans Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListAssociationVersionsla section Référence des AWS CLI commandes.

`list-associations`

L'exemple de code suivant montre comment utiliser`list-associations`.

AWS CLI

Exemple 1 : répertorier vos associations pour une instance spécifique

L'exemple de liste-associations suivant répertorie toutes les associations avec le paramètre Update AssociationName. SSMAgent

```
aws ssm list-associations /
```

```
    --association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

Sortie :

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-016648b75dd622dab"
                    ]
                }
            ],
            "Overview": {
                "Status": "Pending",
                "DetailedStatus": "Associated",
                "AssociationStatusAggregatedCount": {
                    "Pending": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

Pour plus d'informations, consultez Utilisation des associations dans Systems Manager dans le Guide de l'utilisateur Systems Manager.

Exemple 2 : répertorier vos associations pour un document spécifique

L'exemple de list-associations suivant répertorie toutes les associations pour le document spécifié.

```
aws ssm list-associations /
    --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"
```

Sortie :


```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550505828.548,
            "Overview": {
                "Status": "Success",
                "DetailedStatus": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        },
    {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-9876543210abcdef0",
            "AssociationId": "fbc07ef7-b985-4684-b82b-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-9876543210abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550507531.0,
            "Overview": {
                "Status": "Success",
```

```
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez Utilisation des associations dans Systems Manager dans le Guide de l'utilisateur Systems Manager.

- Pour plus de détails sur l'API, voir ListAssociationsla section Référence des AWS CLI commandes.

`list-command-invocations`

L'exemple de code suivant montre comment utiliser`list-command-invocations`.

AWS CLI

Pour répertorier les invocations d'une commande spécifique

L'exemple de `list-command-invocations` suivant répertorie toutes les invocations d'une commande.

```
aws ssm list-command-invocations \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --details
```

Sortie :

```
{
    "CommandInvocations": [
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.089,
            "Status": "Success",
```

```
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.419,
                    "ResponseFinishDateTime": 1582136283.51,
                    "Output": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
 update skipped\n",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": ""
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        },
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-0471e04240EXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.02,
            "Status": "Success",
            "StatusDetails": "Success",
```

```
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.812,
                    "ResponseFinishDateTime": 1582136295.031,
                    "Output": "Updating amazon-ssm-agent from 2.3.672.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/amazon-ssm-agent-updater-
snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/amazon-ssm-agent-snap-amd64.tar.gz
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
amazon-ssm-agent/2.3.842.0/amazon-ssm-agent-snap-amd64.tar.gz\nInitiating amazon-
ssm-agent update to 2.3.842.0\namazon-ssm-agent updated successfully to 2.3.842.0",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/
i-0471e04240EXAMPLE/awsupdateSsmAgent"
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```


Pour plus d'informations, consultez Comprendre les statuts de commande dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListCommandInvocationsla section Référence des AWS CLI commandes.

`list-commands`

L'exemple de code suivant montre comment utiliser`list-commands`.

AWS CLI

Exemple 1 : obtention du statut d’une commande spécifique

L’exemple de `list-commands` suivant récupère et affiche le statut de la commande spécifiée.

```
aws ssm list-commands \
    --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

Exemple 2 : obtention du statut des commandes demandées après une date donnée

L’exemple de `list-commands` suivant permet de récupérer les détails des commandes demandées après la date spécifiée.

```
aws ssm list-commands \
    --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Exemple 3 : Pour répertorier toutes les commandes demandées dans un AWS compte

L'`list-commands`exemple suivant répertorie toutes les commandes demandées par les utilisateurs du AWS compte courant et de la région.

```
aws ssm list-commands
```

Sortie :

```
{
    "Commands": [
        {
            "CommandId": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "ExpiresAfter": "2020-02-19T11:28:02.500000-08:00",
```

```
            "Parameters": {},
            "InstanceIds": [
                "i-028ea792daEXAMPLE",
                "i-02feef8c46EXAMPLE",
                "i-038613f3f0EXAMPLE",
                "i-03a530a2d4EXAMPLE",
                "i-083b678d37EXAMPLE",
                "i-0dee81debaEXAMPLE"
            ],
            "Targets": [],
            "RequestedDateTime": "2020-02-19T10:18:02.500000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "",
            "OutputS3KeyPrefix": "",
            "MaxConcurrency": "50",
            "MaxErrors": "100%",
            "TargetCount": 6,
            "CompletedCount": 6,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
            "CommandId": "e9ade581-c03d-476b-9b07-26667EXAMPLE",
            "DocumentName": "AWS-FindWindowsUpdates",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:37:31.874000-08:00",
            "Parameters": {
                "KbArticleIds": [
                    ""
                ],
                "UpdateLevel": [
                    "All"
```

```
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:27:31.874000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
            "CommandId": "d539b6c3-70e8-4853-80e5-0ce4fEXAMPLE",
            "DocumentName": "AWS-RunPatchBaseline",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:21:04.350000-08:00",
```

```
            "Parameters": {
                "InstallOverrideList": [
                    ""
                ],
                "Operation": [
                    "Install"
                ],
                "RebootOption": [
                    "RebootIfNeeded"
                ],
                "SnapshotId": [
                    ""
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:11:04.350000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
```

```
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListCommandsla section Référence des AWS CLI commandes.

`list-compliance-items`

L'exemple de code suivant montre comment utiliser`list-compliance-items`.

AWS CLI

Pour répertorier les éléments de conformité d'une instance spécifique

Cet exemple répertorie tous les éléments de conformité pour l'instance spécifiée.

Commande :

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance"
```

Sortie :

```
{
  "ComplianceItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "8dfe3659-4309-493a-8755-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
```

```
            "Severity": "UNSPECIFIED",
            "ExecutionSummary": {
                "ExecutionTime": 1550408470.0
            },
            "Details": {
                "DocumentName": "AWS-GatherSoftwareInventory",
                "DocumentVersion": "1"
            }
        },
        {
            "ComplianceType": "Association",
            "ResourceType": "ManagedInstance",
            "ResourceId": "i-1234567890abcdef0",
            "Id": "e4c2ed6d-516f-41aa-aa2a-0123456789ab",
            "Title": "",
            "Status": "COMPLIANT",
            "Severity": "UNSPECIFIED",
            "ExecutionSummary": {
                "ExecutionTime": 1550508475.0
            },
            "Details": {
                "DocumentName": "AWS-UpdateSSMAgent",
                "DocumentVersion": "1"
            }
        },
                ...
    ],
    "NextToken": "--token string truncated--"
}
```

Pour répertorier les éléments de conformité pour une instance et un ID d'association spécifiques

Cet exemple répertorie tous les éléments de conformité pour l'instance et l'ID d'association spécifiés.

Commande :

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ComplianceType,Values=Association,Type=EQUAL" "Key=Id,Values=e4c2ed6d-516f-41aa
aa2a-0123456789ab,Type=EQUAL"
```

Pour répertorier les éléments de conformité d'une instance après une date et heure spécifiques

Cet exemple répertorie tous les éléments de conformité d'une instance après la date et l'heure spécifiées.

Commande :

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

- Pour plus de détails sur l'API, voir ListComplianceItemsla section Référence des AWS CLI commandes.

`list-compliance-summaries`

L'exemple de code suivant montre comment utiliser`list-compliance-summaries`.

AWS CLI

Pour répertorier les récapitulatifs de conformité pour tous les types de conformité

Cet exemple répertorie les récapitulatifs de conformité pour tous les types de conformité de votre compte.

Commande :

```
aws ssm list-compliance-summaries
```

Sortie :

```
{
  "ComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "CompliantSummary": {
              "CompliantCount": 2,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
                  "MediumCount": 0,
                  "LowCount": 0,
                  "InformationalCount": 0,
                  "UnspecifiedCount": 2
```

```
                }
            },
            "NonCompliantSummary": {
                "NonCompliantCount": 0,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 0
                }
            }
        },
        {
            "ComplianceType": "Patch",
            "CompliantSummary": {
                "CompliantCount": 1,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 1
                }
            },
            "NonCompliantSummary": {
                "NonCompliantCount": 1,
                "SeveritySummary": {
                    "CriticalCount": 1,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 0
                }
            }
        },
                ...
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

Pour répertorier les récapitulatifs de conformité pour un type de conformité spécifique

Cet exemple répertorie les récapitulatifs de conformité pour le type de conformité du correctif.

Commande :

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Pour plus de détails sur l'API, voir ListComplianceSummariesla section Référence des AWS CLI commandes.

`list-document-metadata-history`

L'exemple de code suivant montre comment utiliser`list-document-metadata-history`.

AWS CLI

Exemple : pour consulter l'historique des approbations et le statut d'un modèle de modification

L'`list-document-metadata-history`exemple suivant renvoie l'historique des approbations pour le modèle de modification de Change Manager spécifié.

```
aws ssm list-document-metadata-history \
    --name MyChangeManageTemplate \
    --metadata DocumentReviews
```

Sortie :

```
{
    "Name": "MyChangeManagerTemplate",
    "DocumentVersion": "1",
    "Author": "arn:aws:iam::111222333444;:user/JohnDoe",
    "Metadata": {
        "ReviewerResponse": [
            {
                "CreateTime": "2021-07-30T11:58:28.025000-07:00",
                "UpdatedTime": "2021-07-30T12:01:19.274000-07:00",
                "ReviewStatus": "APPROVED",
                "Comment": [
                    {
                        "Type": "COMMENT",
```

```
                        "Content": "I approve this template version"
                    }
                ],
                "Reviewer": "arn:aws:iam::111222333444;:user/ShirleyRodriguez"
            },
            {
                "CreateTime": "2021-07-30T11:58:28.025000-07:00",
                "UpdatedTime": "2021-07-30T11:58:28.025000-07:00",
                "ReviewStatus": "PENDING"
            }
        ]
    }
}
```

Pour plus d'informations, consultez la section Révision et approbation ou rejet des modèles de modification dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListDocumentMetadataHistoryla section Référence des AWS CLI commandes.

`list-document-versions`

L'exemple de code suivant montre comment utiliser`list-document-versions`.

AWS CLI

Pour répertorier les versions d’un document

L’exemple de `list-document-versions` suivant répertorie toutes les versions d’un document Systems Manager.

```
aws ssm list-document-versions \
    --name "Example"
```

Sortie :

```
{
    "DocumentVersions": [
        {
            "Name": "Example",
            "DocumentVersion": "1",
            "CreatedDate": 1583257938.266,
```

```
            "IsDefaultVersion": true,
            "DocumentFormat": "YAML",
            "Status": "Active"
        }
    ]
}
```

Pour plus d'informations, consultez Envoi de commandes qui utilisent le paramètre de version de document dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListDocumentVersionsla section Référence des AWS CLI commandes.

`list-documents`

L'exemple de code suivant montre comment utiliser`list-documents`.

AWS CLI

Exemple 1 : répertorier les documents

L'exemple de `list-documents` suivant répertorie les documents appartenant au compte de la demande étiquetés avec la balise personnalisée.

```
aws ssm list-documents \
    --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Sortie :

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "29884EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Automation",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": [
```

```
                {
                    "Key": "DocUse",
                    "Value": "Testing"
                }
            ]
        }
    ]
}
```

Pour plus d'informations, consultez Documents AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 2 : répertorier les documents partagés

L'`list-documents`exemple suivant répertorie les documents partagés, y compris les documents partagés privés qui ne sont pas détenus par AWS.

```
aws ssm list-documents \
    --filters Key=Name,Values=sharedDocNamePrefix  Key=Owner,Values=Private
```

Sortie :

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "12345EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Command",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": []
        }
    ]
}
```

Pour plus d'informations, consultez Documents AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListDocumentsla section Référence des AWS CLI commandes.

`list-inventory-entries`

L'exemple de code suivant montre comment utiliser`list-inventory-entries`.

AWS CLI

Exemple 1 : affichage des entrées d'un type d'inventaire spécifique pour une instance

L'`list-inventory-entries`exemple suivant répertorie les entrées d'inventaire pour le type:Application AWS inventory sur une instance spécifique.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "AWS:Application"
```

Sortie :

```
{
  "TypeName": "AWS:Application",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.1",
  "CaptureTime": "2019-02-15T12:17:55Z",
  "Entries": [
    {
      "Architecture": "i386",
      "Name": "Amazon SSM Agent",
      "PackageId": "{88a60be2-89a1-4df8-812a-80863c2a2b68}",
      "Publisher": "Amazon Web Services",
      "Version": "2.3.274.0"
    },
    {
      "Architecture": "x86_64",
      "InstalledTime": "2018-05-03T13:42:34Z",
      "Name": "AmazonCloudWatchAgent",
      "Publisher": "",
      "Version": "1.200442.0"
    }
  ]
}
```

Exemple 2 : affichage des entrées d'inventaire personnalisées affectées à une instance

L'exemple de `list-inventory-entries` suivant répertorie une entrée d'inventaire personnalisée affectée à une instance.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "Custom:RackInfo"
```

Sortie :

```
{
  "TypeName": "Custom:RackInfo",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.0",
  "CaptureTime": "2021-05-22T10:01:01Z",
  "Entries": [
    {
      "RackLocation": "Bay B/Row C/Rack D/Shelf E"
    }
  ]
}
```

- Pour plus de détails sur l'API, voir ListInventoryEntriesla section Référence des AWS CLI commandes.

`list-ops-item-related-items`

L'exemple de code suivant montre comment utiliser`list-ops-item-related-items`.

AWS CLI

Pour répertorier les ressources connexes d'un OpsItem

L'`list-ops-item-related-items`exemple suivant répertorie les ressources liées aux éléments connexes d'un. OpsItem

```
aws ssm list-ops-item-related-items \
    --ops-item-id "oi-f99f2EXAMPLE"
```

Sortie :

```
{
    "Summaries": [
        {
            "OpsItemId": "oi-f99f2EXAMPLE",
            "AssociationId": "e2036148-cccb-490e-ac2a-390e5EXAMPLE",
            "ResourceType": "AWS::SSMIncidents::IncidentRecord",
            "AssociationType": "IsParentOf",
            "ResourceUri": "arn:aws:ssm-incidents::111122223333:incident-record/
example-response/64bd9b45-1d0e-2622-840d-03a87a1451fa",
            "CreatedBy": {
                "Arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForIncidentManager/IncidentResponse"
            },
            "CreatedTime": "2021-08-11T18:47:14.994000+00:00",
            "LastModifiedBy": {
                "Arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForIncidentManager/IncidentResponse"
            },
            "LastModifiedTime": "2021-08-11T18:47:14.994000+00:00"
        }
    ]
}
```

Pour plus d'informations, voir Working with Incident Manager incidents OpsCenter dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir ListOpsItemRelatedItemsla section Référence des AWS CLI commandes.

`list-resource-compliance-summaries`

L'exemple de code suivant montre comment utiliser`list-resource-compliance-summaries`.

AWS CLI

Pour répertorier les récapitulatifs de conformité du niveau des ressources

Cet exemple répertorie les récapitulatifs de conformité du niveau des ressources.

Commande :

```
aws ssm list-resource-compliance-summaries
```

Sortie :


```
{
  "ResourceComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Status": "COMPLIANT",
          "OverallSeverity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550509273.0
          },
          "CompliantSummary": {
              "CompliantCount": 2,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
                  "MediumCount": 0,
                  "LowCount": 0,
                  "InformationalCount": 0,
                  "UnspecifiedCount": 2
              }
          },
          "NonCompliantSummary": {
              "NonCompliantCount": 0,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
                  "MediumCount": 0,
                  "LowCount": 0,
                  "InformationalCount": 0,
                  "UnspecifiedCount": 0
              }
          }
      },
      {
          "ComplianceType": "Patch",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-9876543210abcdef0",
          "Status": "COMPLIANT",
          "OverallSeverity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550248550.0,
```

```
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "ExecutionType": "Command"
        },
        "CompliantSummary": {
            "CompliantCount": 397,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 397
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 0,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    }
  ],
  "NextToken": "--token string truncated--"
}
```

Pour répertorier les récapitulatifs de conformité du niveau des ressources pour un type de conformité spécifique

Cet exemple répertorie les récapitulatifs de conformité du niveau des ressources pour le type de conformité du correctif.

Commande :

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Pour plus de détails sur l'API, voir ListResourceComplianceSummariesla section Référence des AWS CLI commandes.

`list-resource-data-sync`

L'exemple de code suivant montre comment utiliser`list-resource-data-sync`.

AWS CLI

Pour répertorier les configurations de synchronisation des données de vos ressources

Cet exemple permet de récupérer des informations sur les configurations de synchronisation des données de vos ressources.

```
aws ssm list-resource-data-sync
```

Sortie :

```
{
    "ResourceDataSyncItems": [
        {
            "SyncName": "MyResourceDataSync",
            "S3Destination": {
                "BucketName": "ssm-resource-data-sync",
                "SyncFormat": "JsonSerDe",
                "Region": "us-east-1"
            },
            "LastSyncTime": 1550261472.003,
            "LastSuccessfulSyncTime": 1550261472.003,
            "LastStatus": "Successful",
            "SyncCreatedTime": 1543235736.72,
            "LastSyncStatusMessage": "The sync was successfully completed"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir ListResourceDataSyncla section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises appliquées à un référentiel de correctifs

L'exemple de `list-tags-for-resource` suivant répertorie les balises appliquées à un référentiel de correctifs.

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0"
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Region",
            "Value": "EMEA"
        }
    ]
}
```

Pour plus d'informations, consultez la section AWS Ressources de balisage dans le manuel de référence AWS général.

- Pour plus de détails sur l'API, voir ListTagsForResourcela section Référence des AWS CLI commandes.

`modify-document-permission`

L'exemple de code suivant montre comment utiliser`modify-document-permission`.

AWS CLI

Pour modifier les autorisations sur les documents

L'exemple de `modify-document-permission` suivant partage publiquement un document Systems Manager.

```
aws ssm modify-document-permission \
    --name "Example" \
    --permission-type "Share" \
    --account-ids-to-add "All"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Partager un document Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir ModifyDocumentPermissionla section Référence des AWS CLI commandes.

`put-compliance-items`

L'exemple de code suivant montre comment utiliser`put-compliance-items`.

AWS CLI

Pour enregistrer un type de conformité et des détails de conformité dans une instance désignée

Cet exemple enregistre le type de conformité `Custom:AVCheck` dans l'instance gérée spécifiée. Il n'y a pas de sortie si la commande réussit.

Commande :

```
aws ssm put-compliance-items --resource-id "i-1234567890abcdef0" --
resource-type "ManagedInstance" --compliance-type "Custom:AVCheck"
 --execution-summary "ExecutionTime=2019-02-18T16:00:00Z" --
items "Id=Version2.0,Title=ScanHost,Severity=CRITICAL,Status=COMPLIANT"
```

- Pour plus de détails sur l'API, voir PutComplianceItemsla section Référence des AWS CLI commandes.

`put-inventory`

L'exemple de code suivant montre comment utiliser`put-inventory`.

AWS CLI

Pour attribuer des métadonnées client à une instance

Cet exemple attribue les informations sur les emplacements des racks à une instance. Il n'y a pas de sortie si la commande réussit.

Commande (Linux) :

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items
 '[{"TypeName": "Custom:RackInfo","SchemaVersion": "1.0","CaptureTime":
 "2019-01-22T10:01:01Z","Content":[{"RackLocation": "Bay B/Row C/Rack D/Shelf
 E"}]}]'
```

Commande (Windows) :

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --
items "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Content=[
 B/Row C/Rack D/Shelf F'}]"
```

- Pour plus de détails sur l'API, voir PutInventorylla section Référence des AWS CLI commandes.

put-parameter

L'exemple de code suivant montre comment utiliserput-parameter.

AWS CLI

Exemple 1 : modification de la valeur d'un paramètre

L'exemple de put-parameter suivant modifie la valeur du paramètre spécifié.

```
aws ssm put-parameter \
    --name "MyStringParameter" \
    --type "String" \
    --value "Vici" \
    --overwrite
```

Sortie :

```
{
    "Version": 2,
    "Tier": "Standard"
}
```

Pour plus d'informations, reportez-vous aux sections Création d'un paramètre Systems Manager (AWS CLI), Gestion des niveaux de paramètres et Utilisation des politiques de paramètres dans le Guide de l'utilisateur de AWS Systems Manager.

Exemple 2 : création d'un paramètre avancé

L'exemple de `put-parameter` suivant crée un paramètre avancé.

```
aws ssm put-parameter \
    --name "MyAdvancedParameter" \
    --description "This is an advanced parameter" \
    --value "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
 tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
 quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat
 [truncated]" \
    --type "String" \
    --tier Advanced
```

Sortie :

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Pour plus d'informations, reportez-vous aux sections Création d'un paramètre Systems Manager (AWS CLI), Gestion des niveaux de paramètres et Utilisation des politiques de paramètres dans le Guide de l'utilisateur de AWS Systems Manager.

Exemple 3 : conversion d'un paramètre standard en paramètre avancé

L'`put-parameter`exemple suivant convertit un paramètre standard existant en paramètre avancé.

```
aws ssm put-parameter \
    --name "MyConvertedParameter" \
    --value "abc123" \
    --type "String" \
    --tier Advanced \
    --overwrite
```

Sortie :

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Pour plus d'informations, reportez-vous aux sections Création d'un paramètre Systems Manager (AWS CLI), Gestion des niveaux de paramètres et Utilisation des politiques de paramètres dans le Guide de l'utilisateur de AWS Systems Manager.

Exemple 4 : création d'un paramètre avec une politique attachée

L'exemple de `put-parameter` suivant crée un paramètre avancé auquel est attachée une politique de paramètre.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification\",
\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type\":
\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",\"Unit
\":\"Days\"}}]"
```

Sortie :

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Pour plus d'informations, reportez-vous aux sections Création d'un paramètre Systems Manager (AWS CLI), Gestion des niveaux de paramètres et Utilisation des politiques de paramètres dans le Guide de l'utilisateur de AWS Systems Manager.

Exemple 5 : ajout d'une politique à un paramètre existant

L'exemple de `put-parameter` suivant attache une politique à un paramètre avancé existant.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
```

```
    --value "N3wP@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification\",
\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type\":
\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",\"Unit
\":\"Days\"}}]"
    --overwrite
```

Sortie :

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Pour plus d'informations, reportez-vous aux sections Création d'un paramètre Systems Manager (AWS CLI), Gestion des niveaux de paramètres et Utilisation des politiques de paramètres dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir PutParameterla section Référence des AWS CLI commandes.

`register-default-patch-baseline`

L'exemple de code suivant montre comment utiliser`register-default-patch-baseline`.

AWS CLI

Pour définir le référentiel de correctifs par défaut

L’exemple de `register-default-patch-baseline` suivant enregistre le référentiel de correctifs personnalisé spécifié en tant que référentiel de correctifs par défaut pour le type de système d’exploitation qu’il prend en charge.

```
aws ssm register-default-patch-baseline \
    --baseline-id "pb-abc123cf9bEXAMPLE"
```

Sortie :

```
{
```

```
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

L'`register-default-patch-baseline`exemple suivant enregistre la ligne de base de correctifs par défaut fournie par AWS CentOS comme ligne de base de correctifs par défaut.

```
aws ssm register-default-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646ed"
```

Sortie :

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

Pour plus d'informations, consultez À propos des référentiels de correctifs prédéfinis et personnalisés dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir RegisterDefaultPatchBaselinela section Référence des AWS CLI commandes.

`register-patch-baseline-for-patch-group`

L'exemple de code suivant montre comment utiliser`register-patch-baseline-for-patch-group`.

AWS CLI

Pour enregistrer un référentiel de correctifs pour un groupe de correctifs

L'exemple de `register-patch-baseline-for-patch-group` suivant permet d'enregistrer un référentiel de correctifs pour un groupe de correctifs.

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-045f10b4f382baeda" \
    --patch-group "Production"
```

Sortie :

```
{
```

```
    "BaselineId": "pb-045f10b4f382baeda",
    "PatchGroup": "Production"
}
```

Pour plus d'informations, consultez les sections Créer un groupe de correctifs < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-group-tagging .html>__ et Ajouter un groupe de correctifs à une ligne de base de correctifs dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir RegisterPatchBaselineForPatchGroupla section Référence des AWS CLI commandes.

`register-target-with-maintenance-window`

L'exemple de code suivant montre comment utiliser`register-target-with-maintenance-window`.

AWS CLI

Exemple 1 : enregistrement d'une cible unique avec une fenêtre de maintenance

L'exemple de `register-target-with-maintenance-window` suivant enregistre une instance avec une fenêtre de maintenance.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \
    --owner-information "Single instance" \
    --resource-type "INSTANCE"
```

Sortie :

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemple 2 : pour enregistrer plusieurs cibles avec une fenêtre de maintenance à l'aide d'une instance IDs

L'`register-target-with-maintenance-window`exemple suivant enregistre deux instances avec une fenêtre de maintenance en spécifiant leur instance IDs.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862,i-0cb2b964d3e14fd9f" \
    --owner-information "Two instances in a list" \
    --resource-type "INSTANCE"
```

Sortie :

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemple 3 : enregistrement de cibles avec une fenêtre de maintenance à l'aide des balises de ressources

L'exemple de `register-target-with-maintenance-window` suivant enregistre des instances avec une fenêtre de maintenance en spécifiant les balises de ressources qui ont été appliquées aux instances.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-06cf17cbefcb4bf4f" \
    --targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \
    --owner-information "Production Web Servers" \
    --resource-type "INSTANCE"
```

Sortie :

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemple 4 : enregistrement de cibles à l'aide d'un groupe de clés de balises

L'exemple de `register-target-with-maintenance-window` suivant enregistre des instances auxquelles une ou plusieurs clés de balises ont été attribuées, quelle que soit la valeur de leur clé.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
```

```
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Sortie :

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Exemple 5 : pour enregistrer des cibles à l'aide d'un nom de groupe de ressources

L'exemple de `register-target-with-maintenance-window` suivant permet d'enregistrer un groupe de ressources spécifié, quel que soit le type de ressources qu'il contient.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Sortie :

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Pour plus d'informations, consultez la section Enregistrer une instance cible avec la fenêtre de maintenance (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir RegisterTargetWithMaintenanceWindowla section Référence des AWS CLI commandes.

`register-task-with-maintenance-window`

L'exemple de code suivant montre comment utiliser`register-task-with-maintenance-window`.

AWS CLI

Exemple 1 : enregistrement d'une tâche d'automatisation avec une fenêtre de maintenance

L'exemple de `register-task-with-maintenance-window` suivant enregistre une tâche d'automatisation avec une fenêtre de maintenance qui est ciblée sur une instance.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649EXAMPLE" \
    --targets Key=InstanceIds,Values=i-1234520122EXAMPLE \
    --task-arn AWS-RestartEC2Instance \
    --service-role-arn arn:aws:iam::111222333444:role/SSM --task-type AUTOMATION \
    --task-invocation-parameters "{\"Automation\":{\"DocumentVersion\":\"\$LATEST\",
\"Parameters\":{\"InstanceId\":[\"{{RESOURCE_ID}}\"]}}}" \
    --priority 0 \
    --max-concurrency 1 \
    --max-errors 1 \
    --name "AutomationExample" \
    --description "Restarting EC2 Instance for maintenance"
```

Sortie :

```
{
    "WindowTaskId":"11144444-5555-6666-7777-88888888"
}
```

Pour plus d'informations, consultez la section Enregistrer une tâche avec la fenêtre de maintenance (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

Exemple 2 : enregistrement d'une tâche Lambda avec une fenêtre de maintenance

L'exemple de `register-task-with-maintenance-window` suivant enregistre une tâche Lambda avec une fenêtre de maintenance ciblée sur une instance.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets Key=InstanceIds,Values=i-12344d305eEXAMPLE \
    --task-arn arn:aws:lambda:us-east-1:111222333444:function:SSMTestLAMBDA \
    --service-role-arn arn:aws:iam::111222333444:role/SSM \
    --task-type LAMBDA \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}","Qualifier":"$LATEST"}}' \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Lambda_Example" \
    --description "My Lambda Example"
```

Sortie :

```
{
    "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

Pour plus d'informations, consultez la section Enregistrer une tâche avec la fenêtre de maintenance (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

Exemple 3 : enregistrement d'une tâche Run Command avec une fenêtre de maintenance

L'exemple de `register-task-with-maintenance-window` suivant enregistre une tâche Run Command avec une fenêtre de maintenance ciblée sur une instance.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/SSM" \
    --task-type "RUN_COMMAND" \
    --name "SSMInstallPowerShellModule" \
    --task-arn "AWS-InstallPowerShellModule" \
    --task-invocation-parameters "{\"RunCommand\":{\"Comment\":\"\",
\"OutputS3BucketName\":\"runcommandlogs\",\"Parameters\":{\"commands\":[\"Get-
Module -ListAvailable\"],\"executionTimeout\":[\"3600\"],\"source\":[\"https:\/
\/gallery.technet.microsoft.com\/EZOut-33ae0fb7\/file\/110351\/1\/EZOut.zip\"],
\"workingDirectory\":[\"\\\\\"]},\"TimeoutSeconds\":600}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

Sortie :

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Pour plus d'informations, consultez la section Enregistrer une tâche avec la fenêtre de maintenance (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

Exemple 4 : enregistrement d'une tâche Step Functions avec une fenêtre de maintenance

L'exemple de `register-task-with-maintenance-window` suivant enregistre une tâche Step Functions avec une fenêtre de maintenance ciblée sur une instance.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-1234d787d6EXAMPLE" \
    --targets Key=WindowTargetIds,Values=12347414-69c3-49f8-95b8-ed2dcEXAMPLE \
    --task-arn arn:aws:states:us-
east-1:111222333444:stateMachine:SSMTestStateMachine \
    --service-role-arn arn:aws:iam::111222333444:role/MaintenanceWindows \
    --task-type STEP_FUNCTIONS \
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}"}}' \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Step_Functions_Example" \
    --description "My Step Functions Example"
```

Sortie :

```
{
    "WindowTaskId":"44444444-5555-6666-7777-88888888"
}
```

Pour plus d'informations, consultez la section Enregistrer une tâche avec la fenêtre de maintenance (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

Exemple 5 : enregistrement d'une tâche à l'aide de l'ID cible d'une fenêtre de maintenance

L'exemple de `register-task-with-maintenance-window` suivant enregistre une tâche à l'aide de l'ID cible d'une fenêtre de maintenance. L'ID cible de la fenêtre de maintenance se trouvait dans la sortie de la commande `aws ssm register-target-with-maintenance-window`. Vous pouvez également le récupérer à partir de la sortie de la commande `aws ssm describe-maintenance-window-targets`.

```
aws ssm register-task-with-maintenance-window \
    --targets "Key=WindowTargetIds,Values=350d44e6-28cc-44e2-951f-4b2c9EXAMPLE" \
    --task-arn "AWS-RunShellScript" \
    --service-role-arn "arn:aws:iam::111222333444:role/MaintenanceWindowsRole" \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-parameters  "{\"commands\":{\"Values\":[\"df\"]}}" \
    --max-concurrency 1 \
    --max-errors 1 \
```

```
    --priority 10
```

Sortie :

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Pour plus d'informations, consultez la section Enregistrer une tâche avec la fenêtre de maintenance (AWS CLI) dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir RegisterTaskWithMaintenanceWindowla section Référence des AWS CLI commandes.

`remove-tags-from-resource`

L'exemple de code suivant montre comment utiliser`remove-tags-from-resource`.

AWS CLI

Pour supprimer une balise d'un référentiel de correctifs

L'exemple de `remove-tags-from-resource` suivant supprime des correctifs d'un référentiel de correctifs.

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0" \
    --tag-keys "Region"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section AWS Ressources de balisage dans le manuel de référence AWS général.

- Pour plus de détails sur l'API, voir RemoveTagsFromResourcela section Référence des AWS CLI commandes.

`reset-service-setting`

L'exemple de code suivant montre comment utiliser`reset-service-setting`.

AWS CLI

Pour réinitialiser le paramètre de service pour le débit du Parameter Store

L'`reset-service-setting`exemple suivant réinitialise le paramètre de service pour le débit du Parameter Store dans la région spécifiée afin de ne plus utiliser de débit accru.

```
aws ssm reset-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled
```

Sortie :

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/parameter-store/high-throughput-enabled",
        "SettingValue": "false",
        "LastModifiedDate": 1555532818.578,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled",
        "Status": "Default"
    }
}
```

Pour plus d'informations, consultez la section Increasing Parameter Store Throughput dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir ResetServiceSettingla section Référence des AWS CLI commandes.

`resume-session`

L'exemple de code suivant montre comment utiliser`resume-session`.

AWS CLI

Pour reprendre une session du gestionnaire de session

Cet `resume-session` exemple reprend une session du gestionnaire de session avec une instance après sa déconnexion. Notez que cette commande interactive nécessite l’installation du plugin Session Manager sur l’ordinateur client qui effectue l’appel.

```
aws ssm resume-session \
    --session-id Mary-Major-07a16060613c408b5
```

Sortie :

```
{
    "SessionId": "Mary-Major-07a16060613c408b5",
    "TokenValue":
 "AAEAAVbTGsaOnyvcUoNGqifbv5r/8lgxuQljCuY8qVcvOnoBAAAAAFxtd3jIXAFUUXGTJ7zF/
AWJPwDviOlF5p3dlAgrqVIVO6IEXhkHLz0/1gXKRKEME71E6TLOplLDJAMZ
+kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wjulPLexO8SHl7g5R/
wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11DOIT2dozOqXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd
+ihfGa6MEJJ97Jmat/a2TspEnOjNn9Mvu5iwXIW2yCvWZrGUj+/
QI5Xr7s1XJBEnSKR54o4fN0GV9RWl0RZsZm1m1ki0JJtiwwgZ",
    "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-
Major-07a16060613c408b5?role=publish_subscribe"
}
```

Pour plus d'informations, voir Installer le plug-in Session Manager pour la AWS CLI dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir ResumeSessionla section Référence des AWS CLI commandes.

`send-automation-signal`

L'exemple de code suivant montre comment utiliser`send-automation-signal`.

AWS CLI

Pour envoyer un signal à une exécution automatique

L'`send-automation-signal`exemple suivant envoie un signal d'approbation à une exécution automatique.

```
aws ssm send-automation-signal \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --signal-type "Approve"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Exécution d'un flux de travail d'automatisation avec des approbateurs dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir SendAutomationSignalla section Référence des AWS CLI commandes.

`send-command`

L'exemple de code suivant montre comment utiliser`send-command`.

AWS CLI

Exemple 1 : exécution d'une commande sur une ou plusieurs instances distantes

L'exemple de `send-command` suivant exécute une commande `echo` sur une instance cible.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters 'commands=["echo HelloWorld"]' \
    --targets "Key=instanceids,Values=i-1234567890abcdef0" \
    --comment "echo HelloWorld"
```

Sortie :

```
{
    "Command": {
        "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "",
        "Comment": "echo HelloWorld",
        "ExpiresAfter": 1550181014.717,
        "Parameters": {
            "commands": [
                "echo HelloWorld"
            ]
        },
        "InstanceIds": [
            "i-0f00f008a2dcbefe2"
        ],
        "Targets": [],
        "RequestedDateTime": 1550173814.717,
        "Status": "Pending",
        "StatusDetails": "Pending",
```

```
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
        "MaxErrors": "0",
        "TargetCount": 1,
        "CompletedCount": 0,
        "ErrorCount": 0,
        "DeliveryTimedOutCount": 0,
        "ServiceRole": "",
        "NotificationConfig": {
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        }
    }
}
```

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 2 : pour obtenir des informations sur l'IP d'une instance

L'exemple de `send-command` suivant récupère les informations sur l'IP d'une instance.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig"
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 3 : pour exécuter une commande sur des instances avec des balises spécifiques

L'exemple de `send-command` suivant exécute une commande sur les instances qui ont la clé de balise « ENV » et la valeur « Dev ».

```
aws ssm send-command \
    --targets "Key=tag:ENV,Values=Dev" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=ifconfig"
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 4 : exécuter une commande qui envoie des notifications SNS

L'exemple de `send-command` suivant exécute une commande qui envoie des notifications SNS pour tous les événements de notification et le type de notification `Command`.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \
    --notification-config "NotificationArn=arn:aws:sns:us-
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 5 : Pour exécuter une commande qui renvoie vers S3 et CloudWatch

L'`send-command`exemple suivant exécute une commande qui affiche les détails de la commande dans un compartiment S3 et dans un groupe de CloudWatch journaux Logs.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --output-s3-bucket-name "s3-bucket-name" \
    --output-s3-key-prefix "runcommand" \
```

```
    --cloud-watch-output-
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 6 : pour exécuter des commandes sur plusieurs instances avec des balises différentes

L'exemple de `send-command` suivant exécute une commande sur des instances avec deux clés et valeurs de balises différentes.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 7 : pour cibler plusieurs instances avec la même clé de balise

L'exemple de `send-command` suivant exécute une commande sur des instances ayant la même clé de balise mais des valeurs différentes.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev,Test
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Exécution de commandes à l'aide de Run Command de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 8 : pour exécuter une commande qui utilise un document partagé

L'exemple de `send-command` suivant exécute un document partagé sur une instance cible.

```
aws ssm send-command \
```

```
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \
    --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Voir l'exemple 1 pour un exemple de sortie.

Pour plus d'informations, consultez Utilisation de documents SSM partagés dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir SendCommandla section Référence des AWS CLI commandes.

`start-associations-once`

L'exemple de code suivant montre comment utiliser`start-associations-once`.

AWS CLI

Pour gérer une association immédiatement et une seule fois

L'`start-associations-once`exemple suivant exécute l'association spécifiée immédiatement et une seule fois. Il n'y a pas de sortie si la commande réussit.

```
aws ssm start-associations-once \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Afficher l'historique des associations dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir StartAssociationsOncela section Référence des AWS CLI commandes.

`start-automation-execution`

L'exemple de code suivant montre comment utiliser`start-automation-execution`.

AWS CLI

Exemple 1 : exécution d'un document d'automatisation

L'exemple de `start-automation-execution` suivant exécute un document d'automatisation.

```
aws ssm start-automation-execution \
    --document-name "AWS-UpdateLinuxAmi" \
    --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Sortie :

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Pour plus d'informations, consultez Exécution manuelle d'un flux de travail d'automatisation dans le Guide de l'utilisateur AWS Systems Manager.

Exemple 2 : exécution d'un document d'automatisation partagé

L'exemple de `start-automation-execution` suivant exécute un document d'automatisation partagé.

```
aws ssm start-automation-execution \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Sortie :

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Pour plus d'informations, consultez Utilisation de documents SSM partagés dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, voir StartAutomationExecutionla section Référence des AWS CLI commandes.

`start-change-request-execution`

L'exemple de code suivant montre comment utiliser`start-change-request-execution`.

AWS CLI

Exemple 1 : pour lancer une demande de modification

L'`start-change-request-execution`exemple suivant lance une demande de modification avec un minimum d'options spécifiées.

```
aws ssm start-change-request-execution \
    --change-request-name MyChangeRequest \
    --document-name AWS-HelloWorldChangeTemplate \
    --runbooks '[{"DocumentName": "AWS-HelloWorld","Parameters":
 {"AutomationAssumeRole": ["arn:aws:iam:us-east-2:1112223233444:role/
MyChangeManagerAssumeRole"]}}]' \
    --parameters
 Approver="JohnDoe",ApproverType="IamUser",ApproverSnsTopicArn="arn:aws:sns:us-
east-2:1112223233444:MyNotificationTopic"
```

Sortie :

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

Exemple 2 : Pour démarrer une demande de modification à l'aide d'un fichier JSON externe

L'`start-automation-execution`exemple suivant lance une demande de modification avec plusieurs options spécifiées dans un fichier JSON.

```
aws ssm start-change-request-execution \
    --cli-input-json file://MyChangeRequest.json
```

Contenu de `MyChangeRequest.json` :

```
{
    "ChangeRequestName": "MyChangeRequest",
    "DocumentName": "AWS-HelloWorldChangeTemplate",
    "DocumentVersion": "$DEFAULT",
    "ScheduledTime": "2021-12-30T03:00:00",
    "ScheduledEndTime": "2021-12-30T03:05:00",
    "Tags": [
        {
            "Key": "Purpose",
            "Value": "Testing"
        }
    ],
```

```
    "Parameters": {
        "Approver": [
            "JohnDoe"
        ],
        "ApproverType": [
            "IamUser"
        ],
        "ApproverSnsTopicArn": [
            "arn:aws:sns:us-east-2:111222333444;:MyNotificationTopic
        ]
    },
    "Runbooks": [
        {
            "DocumentName": "AWS-HelloWorld",
            "DocumentVersion": "1",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Parameters": {
                "AutomationAssumeRole": [
                    "arn:aws:iam::111222333444:role/MyChangeManagerAssumeRole"
                ]
            }
        }
    ],
    "ChangeDetails": "### Document Name: HelloWorldChangeTemplate\n\n## What does
 this document do?\nThis change template demonstrates the feature set available
 for creating change templates for Change Manager. This template starts a Runbook
 workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters
\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
 approvers.\n* Approver: (Required) The name of the approver to send this request
 to.\n* ApproverType: (Required) The type of reviewer.\n  * Allowed Values: IamUser,
 IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\nThis document has no
 outputs \n"
}
```

Sortie :

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

Pour plus d'informations, consultez la section Création de demandes de modification dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, voir StartChangeRequestExecutionla section Référence des AWS CLI commandes.

`start-session`

L'exemple de code suivant montre comment utiliser`start-session`.

AWS CLI

Exemple 1 : démarrage d'une session Session Manager

Cet exemple de `start-session` établit une connexion avec une instance pour une session Session Manager. Notez que cette commande interactive nécessite l'installation du plugin Session Manager sur l'ordinateur client qui effectue l'appel.

```
aws ssm start-session \
    --target "i-1234567890abcdef0"
```

Sortie :

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

Exemple 2 : démarrage d'une session Session Manager en utilisant SSH

Cet exemple de `start-session` établit une connexion avec une instance pour une session Session Manager en utilisant SSH. Notez que cette commande interactive nécessite que le plug-in Session Manager soit installé sur la machine cliente effectuant l'appel, et que la commande utilise l'utilisateur par défaut sur l'instance, comme `ec2-user` pour les EC2 instances pour Linux.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

Sortie :

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

Pour plus d'informations, consultez Démarrer une session et installer le plug-in Session Manager pour la AWS CLI dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous StartSessionà la section Référence des AWS CLI commandes.

`stop-automation-execution`

L'exemple de code suivant montre comment utiliser`stop-automation-execution`.

AWS CLI

Pour arrêter l’exécution d’une automatisation

L’exemple de `stop-automation-execution` suivant arrête un document d’automatisation.

```
aws ssm stop-automation-execution
    --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Cette commande ne produit aucun résultat.

Pour plus d’informations, consultez Exécution manuelle d’un flux de travail d’automatisation dans le Guide de l’utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous StopAutomationExecutionà la section Référence des AWS CLI commandes.

`terminate-session`

L'exemple de code suivant montre comment utiliser`terminate-session`.

AWS CLI

Pour mettre fin à une session du gestionnaire de session

Cet `terminate-session` exemple met définitivement fin à une session créée par l'utilisateur « Shirley-Rodriguez » et ferme la connexion de données entre le client Session Manager et l'agent SSM sur l'instance.

```
aws ssm terminate-session \
    --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

Sortie :

```
{
    "SessionId": "Shirley-Rodriguez-07a16060613c408b5"
}
```

Pour plus d'informations, consultez la section Mettre fin à une session dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous TerminateSessionà la section Référence des AWS CLI commandes.

`unlabel-parameter-version`

L'exemple de code suivant montre comment utiliser`unlabel-parameter-version`.

AWS CLI

Pour supprimer des libellés de paramètres

L'`unlabel-parameter-version`exemple suivant supprime les étiquettes spécifiées de la version de paramètre donnée.

```
aws ssm unlabel-parameter-version \
    --name "parameterName" \
    --parameter-version "version" \
    --labels "label_1" "label_2" "label_3"
```

Sortie :

```
{
    "RemovedLabels": [
        "label_1"
        "label_2"
        "label_3"
    ],
    "InvalidLabels": []
}
```

Pour plus d'informations, consultez la section Supprimer les étiquettes de paramètres (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UnlabelParameterVersionà la section Référence des AWS CLI commandes.

`update-association-status`

L'exemple de code suivant montre comment utiliser`update-association-status`.

AWS CLI

Pour mettre à jour le statut de l’association

L’exemple de `update-association-status` suivant met à jour le statut de l’association entre une instance et un document.

```
aws ssm update-association-status \
    --name "AWS-UpdateSSMAgent" \
    --instance-id "i-1234567890abcdef0" \
    --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additiona
Config-Needed"
```

Sortie :

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "AssociationVersion": "1",
        "Date": 1550507529.604,
        "LastUpdateAssociationDate": 1550507806.974,
        "Status": {
            "Date": 1424421071.0,
            "Name": "Pending",
            "Message": "temp_status_change",
            "AdditionalInfo": "Additional-Config-Needed"
        },
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
                "Success": 1
            }
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-1234567890abcdef0"
                ]
```

```
            }
        ],
        "LastExecutionDate": 1550507808.0,
        "LastSuccessfulExecutionDate": 1550507808.0
    }
}
```

Pour plus d'informations, consultez Utilisation des associations dans Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateAssociationStatusà la section Référence des AWS CLI commandes.

`update-association`

L'exemple de code suivant montre comment utiliser`update-association`.

AWS CLI

Exemple 1 : mise à jour d'une association de documents

L'exemple de `update-association` suivant met à jour une association avec une nouvelle version de document.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --document-version "\$LATEST"
```

Sortie :

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "AssociationVersion": "2",
        "Date": 1550508093.293,
        "LastUpdateAssociationDate": 1550508106.596,
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$LATEST",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
```

```
        "Targets": [
            {
                "Key": "tag:Name",
                "Values": [
                    "Linux"
                ]
            }
        ],
        "LastExecutionDate": 1550508094.879,
        "LastSuccessfulExecutionDate": 1550508094.879
    }
}
```

Pour plus d'informations, consultez Modifier et créer une nouvelle version d'une association (français non garanti) dans le AWS Guide de l'utilisateur Systems Manager (français non garanti).

Exemple 2 : mise à jour de l'expression de l'horaire d'une association

L'exemple de `update-association` suivant met à jour l'expression de l'horaire pour l'association spécifiée.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

Sortie :

```
{
    "AssociationDescription": {
        "Name": "AWS-HelloWorld",
        "AssociationVersion": "2",
        "Date": "2021-02-08T13:54:19.203000-08:00",
        "LastUpdateAssociationDate": "2021-06-29T11:51:07.933000-07:00",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "aws:NoOpAutomationTag",
                "Values": [
```

```
                    "AWS-NoOpAutomationTarget-Value"
                ]
            }
        ],
        "ScheduleExpression": "cron(0 0 0/4 1/1 * ? *)",
        "LastExecutionDate": "2021-06-26T19:00:48.110000-07:00",
        "ApplyOnlyAtCronInterval": false
    }
}
```

Pour plus d'informations, consultez Modifier et créer une nouvelle version d'une association (français non garanti) dans le AWS Guide de l'utilisateur Systems Manager (français non garanti).

- Pour plus de détails sur l'API, reportez-vous UpdateAssociationà la section Référence des AWS CLI commandes.

`update-document-default-version`

L'exemple de code suivant montre comment utiliser`update-document-default-version`.

AWS CLI

Pour mettre à jour la version par défaut d'un document

L'exemple de `update-document-default-version` suivant met à jour la version par défaut d'un document Systems Manager.

```
aws ssm update-document-default-version \
    --name "Example" \
    --document-version "2"
```

Sortie :

```
{
    "Description": {
        "Name": "Example",
        "DefaultVersion": "2"
    }
}
```

Pour plus d'informations, consultez Écrire le contenu d'un document SSM dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateDocumentDefaultVersionà la section Référence des AWS CLI commandes.

`update-document-metadata`

L'exemple de code suivant montre comment utiliser`update-document-metadata`.

AWS CLI

Exemple : pour approuver la dernière version d'un modèle de modification

`update-document-metadata`Ce qui suit fournit une approbation pour la dernière version d'un modèle de modification qui a été soumis pour révision.

```
aws ssm update-document-metadata \
    --name MyChangeManagerTemplate \
    --document-reviews 'Action=Approve,Comment=[{Type=Comment,Content=Approved!}]'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Révision et approbation ou rejet des modèles de modification dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateDocumentMetadataà la section Référence des AWS CLI commandes.

`update-document`

L'exemple de code suivant montre comment utiliser`update-document`.

AWS CLI

Pour créer une nouvelle version d’un document

L’exemple de `update-document` suivant crée une nouvelle version d’un document lorsqu’il est exécuté dans un ordinateur Windows. Le document spécifié par `--document` doit être au format JSON. Notez que `file://` doit être référencé et suivi du chemin d’accès au fichier de contenu. En raison du $ au début du paramètre `--document-version`, vous devez entourer la valeur de guillemets doubles sous Windows. Sous Linux, macOS ou lorsque vous y êtes PowerShell invité, vous devez placer la valeur entre guillemets simples.

Version Windows :

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version "$LATEST"
```

Version Linux/Mac :

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version '$LATEST'
```

Sortie :

```
{
  "DocumentDescription": {
      "Status": "Updating",
      "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",
      "Name": "RunShellScript",
      "Parameters": [
          {
              "Type": "StringList",
              "Name": "commands",
              "Description": "(Required) Specify a shell script or a command to
 run."
          }
      ],
      "DocumentType": "Command",
      "PlatformTypes": [
          "Linux"
      ],
      "DocumentVersion": "2",
      "HashType": "Sha256",
      "CreatedDate": 1487899655.152,
      "Owner": "809632081692",
      "SchemaVersion": "2.0",
      "DefaultVersion": "1",
      "LatestVersion": "2",
      "Description": "Run an updated script"
  }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateDocumentà la section Référence des AWS CLI commandes.

`update-maintenance-window-target`

L'exemple de code suivant montre comment utiliser`update-maintenance-window-target`.

AWS CLI

Pour mettre à jour la cible d'une fenêtre de maintenance

L'`update-maintenance-window-target`exemple suivant met à jour uniquement le nom d'une cible de fenêtre de maintenance.

```
aws ssm update-maintenance-window-target \
    --window-id "mw-0c5ed765acEXAMPLE" \
    --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
    --name "NewName" \
    --no-replace
```

Sortie :

```
{
    "Description": "",
    "OwnerInformation": "",
    "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "Targets": [
        {
            "Values": [
                "i-1234567890EXAMPLE"
            ],
            "Key": "InstanceIds"
        }
    ],
    "Name": "NewName"
}
```

Pour plus d'informations, voir Mettre à jour une fenêtre de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateMaintenanceWindowTargetà la section Référence des AWS CLI commandes.

`update-maintenance-window-task`

L'exemple de code suivant montre comment utiliser`update-maintenance-window-task`.

AWS CLI

Pour mettre à jour une tâche de fenêtre de maintenance

L'`update-maintenance-window-task`exemple suivant met à jour le rôle de service pour une tâche de fenêtre de maintenance.

```
aws ssm update-maintenance-window-task \
    --window-id "mw-0c5ed765acEXAMPLE" \
    --window-task-id "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM"
```

Sortie :

```
{
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "MaxErrors": "1",
    "TaskArn": "AWS-UpdateEC2Config",
    "MaxConcurrency": "1",
    "WindowTaskId": "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE",
    "TaskParameters": {},
    "Priority": 1,
    "TaskInvocationParameters": {
        "RunCommand": {
            "TimeoutSeconds": 600,
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ]
            }
        }
    },
    "WindowId": "mw-0c5ed765acEXAMPLE",
```

```
    "Description": "UpdateEC2Config",
    "Targets": [
        {
            "Values": [
                "57e8344e-fe64-4023-8191-6bf05EXAMPLE"
            ],
            "Key": "WindowTargetIds"
        }
    ],
    "Name": "UpdateEC2Config"
}
```

Pour plus d'informations, voir Mettre à jour une fenêtre de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateMaintenanceWindowTaskà la section Référence des AWS CLI commandes.

`update-maintenance-window`

L'exemple de code suivant montre comment utiliser`update-maintenance-window`.

AWS CLI

Exemple 1 : mise à jour d'une fenêtre de maintenance

L'exemple de `update-maintenance-window` suivant met à jour le nom d'une fenêtre de maintenance.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --name "My-Renamed-MW"
```

Sortie :

```
{
    "Cutoff": 1,
    "Name": "My-Renamed-MW",
    "Schedule": "cron(0 16 ? * TUE *)",
    "Enabled": true,
    "AllowUnassociatedTargets": true,
    "WindowId": "mw-1a2b3c4d5e6f7g8h9",
```

```
    "Duration": 4
}
```

Exemple 2 : désactivation d'une fenêtre de maintenance

L'exemple de `update-maintenance-window` suivant désactive une fenêtre de maintenance.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --no-enabled
```

Exemple 3 : activation d'une fenêtre de maintenance

L'exemple de `update-maintenance-window` suivant permet d'activer une fenêtre de maintenance.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --enabled
```

Pour plus d'informations, voir Mettre à jour une fenêtre de maintenance (AWS CLI) dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateMaintenanceWindowà la section Référence des AWS CLI commandes.

`update-managed-instance-role`

L'exemple de code suivant montre comment utiliser`update-managed-instance-role`.

AWS CLI

Pour mettre à jour le rôle IAM d'une instance gérée

L'exemple de `update-managed-instance-role` suivant met à jour le profil d'instance IAM d'une instance gérée.

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
    --iam-role "ExampleRole"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Étape 4 : créer un profil d'instance IAM pour Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateManagedInstanceRoleà la section Référence des AWS CLI commandes.

`update-ops-item`

L'exemple de code suivant montre comment utiliser`update-ops-item`.

AWS CLI

Pour mettre à jour un OpsItem

L'`update-ops-item`exemple suivant met à jour la description, la priorité et la catégorie d'un OpsItem. En outre, la commande spécifie une rubrique SNS dans laquelle les notifications sont envoyées lorsqu'elle OpsItem est modifiée ou modifiée.

```
aws ssm update-ops-item \
    --ops-item-id "oi-287b5EXAMPLE" \
    --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \
    --priority 2 \
    --category "Security" \
    --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

Sortie :

```
This command produces no output.
```

Pour plus d'informations, reportez-vous à la section Travailler avec OpsItems dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateOpsItemà la section Référence des AWS CLI commandes.

`update-patch-baseline`

L'exemple de code suivant montre comment utiliser`update-patch-baseline`.

AWS CLI

Exemple 1 : mise à jour d'un référentiel de correctifs

L'exemple de `update-patch-baseline` suivant ajoute les deux correctifs spécifiés comme rejetés et un correctif comme approuvé au référentiel de correctifs spécifié.

```
aws ssm update-patch-baseline \
        --baseline-id "pb-0123456789abcdef0" \
        --rejected-patches "KB2032276" "MS10-048" \
        --approved-patches "KB2124261"
```

Sortie :

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [
        "KB2124261"
    ],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [
        "KB2032276",
```

```
        "MS10-048"
    ],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

Exemple 2 : renommer un référentiel de correctifs

L'exemple de `update-patch-baseline` suivant renomme le référentiel de correctifs spécifié.

```
aws ssm update-patch-baseline \
    --baseline-id "pb-0713accee01234567" \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Pour plus d'informations, voir Mettre à jour ou supprimer une ligne de base de correctifs < https://docs.aws.amazon.com/systems-manager/ latest/userguide/patch - baseline-update-or-delete .html>`__ dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdatePatchBaselineà la section Référence des AWS CLI commandes.

update-resource-data-sync

L'exemple de code suivant montre comment utiliser`update-resource-data-sync`.

AWS CLI

Pour mettre à jour la synchronisation des données d'une ressource

L'`update-resource-data-sync`exemple suivant met à jour la synchronisation SyncFromSource des données d'une ressource.

```
aws ssm update-resource-data-sync \
    --sync-name exampleSync \
    --sync-type SyncFromSource \
    --sync-source '{"SourceType":"SingleAccountMultiRegions", "SourceRegions":["us-east-1", "us-west-2"]}'
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, reportez-vous à la section Configuration de Systems Manager Explorer pour afficher les données de plusieurs comptes et régions dans le Guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateResourceDataSyncà la section Référence des AWS CLI commandes.

`update-service-setting`

L'exemple de code suivant montre comment utiliser`update-service-setting`.

AWS CLI

Pour mettre à jour le paramètre de service pour le débit du Parameter Store

L'`update-service-setting`exemple suivant met à jour le paramètre de service actuel pour le débit du Parameter Store dans la région spécifiée afin d'utiliser un débit accru.

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled \
    --setting-value true
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Increasing Parameter Store Throughput dans le guide de l'utilisateur de AWS Systems Manager.

- Pour plus de détails sur l'API, reportez-vous UpdateServiceSettingà la section Référence des AWS CLI commandes.

Exemples d'Amazon Textract utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Textract.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`analyze-document`

L'exemple de code suivant montre comment utiliser`analyze-document`.

AWS CLI

Pour analyser le texte d'un document

L'`analyze-document`exemple suivant montre comment analyser le texte d'un document.

Linux/macOS :

```
aws textract analyze-document \
    --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
    --feature-types '["TABLES","FORMS"]'
```

Windows :

```
aws textract analyze-document \
    --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}" \
    --feature-types "[\"TABLES\",\"FORMS\"]" \
    --region region-name
```

Sortie :

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
```

```
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "87586964-d50d-43e2-ace5-8a890657b9a0",
                        "a1e72126-21d9-44f4-a8d6-5c385f9002ba",
                        "e889d012-8a6b-4d2e-b7cd-7a8b327d876a"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "c2227f12-b25d-4e1f-baea-1ee180d926b2"
        }
    ],
    "DocumentMetadata": {
        "Pages": 1
    }
}
```


Pour plus d'informations, consultez Analyser le texte d'un document avec Amazon Textract dans le manuel Amazon Textract Developers Guide

- Pour plus de détails sur l'API, voir AnalyzeDocumentla section Référence des AWS CLI commandes.

`detect-document-text`

L'exemple de code suivant montre comment utiliser`detect-document-text`.

AWS CLI

Pour détecter du texte dans un document

L'exemple suivant montre comment détecter du texte dans un document. `detect-document-text`

Linux/macOS :

```
aws textract detect-document-text \
    --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}'
```

Windows :

```
aws textract detect-document-text \
    --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}" \
    --region region-name
```

Sortie :

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
```

```
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "896a9f10-9e70-4412-81ce-49ead73ed881",
                        "0da18623-dc4c-463d-a3d1-9ac050e9e720",
                        "167338d7-d38c-4760-91f1-79a8ec457bb2"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "21f0535e-60d5-4bc7-adf2-c05dd851fa25"
        },
        {
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "62490c26-37ea-49fa-8034-7a9ff9369c9c",
                        "1e4f3f21-05bd-4da9-ba10-15d01e66604c"
                    ]
                }
            ],
            "Confidence": 89.11581420898438,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33642634749412537,
                    "Top": 0.17169663310050964,
                    "Left": 0.13885067403316498,
                    "Height": 0.49159330129623413
                },
                "Polygon": [
                    {
                        "Y": 0.17169663310050964,
```

```
                        "X": 0.13885067403316498
                    },
                    {
                        "Y": 0.17169663310050964,
                        "X": 0.47527703642845154
                    },
                    {
                        "Y": 0.6632899641990662,
                        "X": 0.47527703642845154
                    },
                    {
                        "Y": 0.6632899641990662,
                        "X": 0.13885067403316498
                    }
                ]
            },
            "Text": "He llo,",
            "BlockType": "LINE",
            "Id": "896a9f10-9e70-4412-81ce-49ead73ed881"
        },
        {
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "19b28058-9516-4352-b929-64d7cef29daf"
                    ]
                }
            ],
            "Confidence": 85.5694351196289,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33182239532470703,
                    "Top": 0.23131252825260162,
                    "Left": 0.5091826915740967,
                    "Height": 0.3766750991344452
                },
                "Polygon": [
                    {
                        "Y": 0.23131252825260162,
                        "X": 0.5091826915740967
                    },
                    {
                        "Y": 0.23131252825260162,
```

```
                        "X": 0.8410050868988037
                    },
                    {
                        "Y": 0.607987642288208,
                        "X": 0.8410050868988037
                    },
                    {
                        "Y": 0.607987642288208,
                        "X": 0.5091826915740967
                    }
                ]
            },
            "Text": "worlc",
            "BlockType": "LINE",
            "Id": "0da18623-dc4c-463d-a3d1-9ac050e9e720"
        }
    ],
    "DocumentMetadata": {
        "Pages": 1
    }
}
```

Pour plus d'informations, consultez la section Détection du texte d'un document avec Amazon Textract dans le manuel Amazon Textract Developers Guide.

- Pour plus de détails sur l'API, voir DetectDocumentTextla section Référence des AWS CLI commandes.

`get-document-analysis`

L'exemple de code suivant montre comment utiliser`get-document-analysis`.

AWS CLI

Pour obtenir les résultats de l'analyse asynchrone du texte d'un document de plusieurs pages

L'`get-document-analysis`exemple suivant montre comment obtenir les résultats de l'analyse de texte asynchrone d'un document de plusieurs pages.

```
aws textract get-document-analysis \
    --job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \
    --max-results 1000
```

Sortie :


```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "75966e64-81c2-4540-9649-d66ec341cd8f",
                        "bb099c24-8282-464c-a179-8a9fa0a057f0",
                        "5ebf522d-f9e4-4dc7-bfae-a288dc094595"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "247c28ee-b63d-4aeb-9af0-5f7ea8ba109e",
            "Page": 1
```

```
        }
    ],
    "NextToken": "cY1W3eTFvoB0cH7YrKVudI4Gb0H8J0xAYLo8xI/JunCIPWCthaKQ+07n/
ElyutsSy0+1VOImoTRmP1zw4P0RFtaeV9Bzhnfedpx1YqwB4xaGDA==",
    "DocumentMetadata": {
        "Pages": 1
    },
    "JobStatus": "SUCCEEDED"
}
```

Pour plus d'informations, consultez la section Détection et analyse du texte dans les documents de plusieurs pages du manuel Amazon Textract Developers Guide

- Pour plus de détails sur l'API, voir GetDocumentAnalysisla section Référence des AWS CLI commandes.

`get-document-text-detection`

L'exemple de code suivant montre comment utiliser`get-document-text-detection`.

AWS CLI

Pour obtenir les résultats de la détection de texte asynchrone dans un document de plusieurs pages

L'`get-document-text-detection`exemple suivant montre comment obtenir les résultats de la détection de texte asynchrone dans un document de plusieurs pages.

```
aws textract get-document-text-detection \
    --job-id 57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9 \
    --max-results 1000
```

Sortie

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
```

```
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "1b926a34-0357-407b-ac8f-ec473160c6a9",
                        "0c35dc17-3605-4c9d-af1a-d9451059df51",
                        "dea3db8a-52c2-41c0-b50c-81f66f4aa758"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "84671a5e-8c99-43be-a9d1-6838965da33e",
            "Page": 1
        }
    ],
    "NextToken": "GcqyoAJuZwujOT35EN4LCI3EUzMtiLq3nKyFFHvU5q1SaIdEBcSty+njNgoWwuMP/
muqc96S4o5NzDqehhXvhkodMyVO5OJGyms5lsrCxibWJw==",
    "DocumentMetadata": {
        "Pages": 1
    },
    "JobStatus": "SUCCEEDED"
}
```

Pour plus d'informations, consultez la section Détection et analyse du texte dans les documents de plusieurs pages du manuel Amazon Textract Developers Guide

- Pour plus de détails sur l'API, voir GetDocumentTextDetectionla section Référence des AWS CLI commandes.

`start-document-analysis`

L'exemple de code suivant montre comment utiliser`start-document-analysis`.

AWS CLI

Pour commencer à analyser le texte d'un document de plusieurs pages

L'`start-document-analysis`exemple suivant montre comment démarrer une analyse asynchrone du texte dans un document de plusieurs pages.

Linux/macOS :

```
aws textract start-document-analysis \
    --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
    --feature-types '["TABLES","FORMS"]' \
    --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Windows :

```
aws textract start-document-analysis \
    --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document
\"}}" \
    --feature-types "[\"TABLES\", \"FORMS\"]" \
    --region region-name \
    --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Sortie :

```
{
    "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"
}
```

Pour plus d'informations, consultez la section Détection et analyse du texte dans les documents de plusieurs pages du manuel Amazon Textract Developers Guide

- Pour plus de détails sur l'API, voir StartDocumentAnalysisla section Référence des AWS CLI commandes.

`start-document-text-detection`

L'exemple de code suivant montre comment utiliser`start-document-text-detection`.

AWS CLI

Pour commencer à détecter du texte dans un document de plusieurs pages

L'`start-document-text-detection`exemple suivant montre comment démarrer la détection asynchrone du texte dans un document de plusieurs pages.

Linux/macOS :

```
aws textract start-document-text-detection \
        --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
        --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleARN"
```

Windows :

```
aws textract start-document-text-detection \
    --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document
\"}}" \
    --region region-name \
    --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Sortie :

```
{
    "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"
}
```

Pour plus d'informations, consultez la section Détection et analyse du texte dans les documents de plusieurs pages du manuel Amazon Textract Developers Guide

- Pour plus de détails sur l'API, voir StartDocumentTextDetectionla section Référence des AWS CLI commandes.

Exemples d'Amazon Transcribe utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Transcribe.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-language-model`

L'exemple de code suivant montre comment utiliser`create-language-model`.

AWS CLI

Exemple 1 : créer un modèle de langage personnalisé en utilisant à la fois des données d'entraînement et de réglage.

L'`create-language-model`exemple suivant crée un modèle de langage personnalisé. Vous pouvez utiliser un modèle de langage personnalisé pour améliorer les performances de transcription dans des domaines tels que le droit, l'hôtellerie, les finances et les assurances. Pour le code de langue, entrez un code de langue valide. Pour base-model-name, spécifiez le modèle de base le mieux adapté à la fréquence d'échantillonnage de l'audio que vous souhaitez transcrire avec votre modèle de langue personnalisé. Pour model-name, spécifiez le nom que vous souhaitez appeler le modèle de langage personnalisé.

```
aws transcribe create-language-model \
    --language-code language-code \
    --base-model-name base-model-name \
    --model-name cli-clm-example \
    --input-data-config S3Uri="s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-for-
training-data",TuningDataS3Uri="s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-for-
```

```
tuning-data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-to-create-a-custom-language-model"
```

Sortie :

```
{
    "LanguageCode": "language-code",
    "BaseModelName": "base-model-name",
    "ModelName": "cli-clm-example",
    "InputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
        "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
        "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-create-a-custom-language-model"
    },
    "ModelStatus": "IN_PROGRESS"
}
```

Pour plus d'informations, consultez Amélioration de la précision de la transcription spécifique à un domaine grâce à des modèles de langue personnalisés dans le Guide du développeur Amazon Transcribe.

Exemple 2 : créer un modèle de langage personnalisé en utilisant uniquement les données d'entraînement.

L'exemple `create-language-model` suivant transcrit un fichier audio. Vous pouvez utiliser un modèle de langage personnalisé pour améliorer les performances de transcription dans des domaines tels que le droit, l'hôtellerie, les finances et les assurances. Pour le code de langue, entrez un code de langue valide. Pour base-model-name, spécifiez le modèle de base le mieux adapté à la fréquence d'échantillonnage de l'audio que vous souhaitez transcrire avec votre modèle de langue personnalisé. Pour model-name, spécifiez le nom que vous souhaitez appeler le modèle de langage personnalisé.

```
aws transcribe create-language-model \
    --language-code en-US \
    --base-model-name base-model-name \
    --model-name cli-clm-example \
    --input-data-config S3Uri="s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-For-
Training-Data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
```

Sortie :

```
{
    "LanguageCode": "en-US",
    "BaseModelName": "base-model-name",
    "ModelName": "cli-clm-example",
    "InputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-For-Training-Data/",
        "DataAccessRoleArn": "arn:aws:iam::your-AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
    },
    "ModelStatus": "IN_PROGRESS"
}
```

Pour plus d'informations, consultez Amélioration de la précision de la transcription spécifique à un domaine grâce à des modèles de langue personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous CreateLanguageModelà la section Référence des AWS CLI commandes.

`create-medical-vocabulary`

L'exemple de code suivant montre comment utiliser`create-medical-vocabulary`.

AWS CLI

Pour créer un vocabulaire médical personnalisé

L'exemple `create-medical-vocabulary` suivant crée un vocabulaire personnalisé. Pour créer un vocabulaire personnalisé, vous devez avoir créé un fichier texte contenant tous les termes que vous souhaitez transcrire de manière plus précise. Pour vocabulary-file-uri, spécifiez l'URI Amazon Simple Storage Service (Amazon S3) de ce fichier texte. Pour le code de langue, spécifiez un code de langue correspondant à la langue du vocabulaire personnalisé. Pour le nom de vocabulaire, spécifiez le nom que vous souhaitez donner au vocabulaire personnalisé.

```
aws transcribe create-medical-vocabulary \
    --vocabulary-name cli-medical-vocab-example \
    --language-code language-code \
    --vocabulary-file-uri https://amzn-s3-demo-bucket.AWS-Region.amazonaws.com/the-
text-file-for-the-medical-custom-vocabulary.txt
```

Sortie :

```
{
    "VocabularyName": "cli-medical-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés médicaux dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous CreateMedicalVocabularyà la section Référence des AWS CLI commandes.

`create-vocabulary-filter`

L'exemple de code suivant montre comment utiliser`create-vocabulary-filter`.

AWS CLI

Pour créer un filtre de vocabulaire

L'`create-vocabulary-filter`exemple suivant crée un filtre de vocabulaire qui utilise un fichier texte contenant une liste de mots que vous ne souhaitez pas voir apparaître dans une transcription. Pour le code de langue, spécifiez le code de langue correspondant à la langue de votre filtre de vocabulaire. Pour vocabulary-filter-file-uri, spécifiez l'URI Amazon Simple Storage Service (Amazon S3) du fichier texte. Pour vocabulary-filter-name, spécifiez le nom de votre filtre de vocabulaire.

```
aws transcribe create-vocabulary-filter \
    --language-code language-code \
    --vocabulary-filter-file-uri s3://amzn-s3-demo-bucket/vocabulary-filter.txt \
    --vocabulary-filter-name cli-vocabulary-filter-example
```

Sortie :

```
{
    "VocabularyFilterName": "cli-vocabulary-filter-example",
    "LanguageCode": "language-code"
}
```

Pour plus d'informations, consultez la section Filtrer les mots indésirables dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateVocabularyFilterà la section Référence des AWS CLI commandes.

`create-vocabulary`

L'exemple de code suivant montre comment utiliser`create-vocabulary`.

AWS CLI

Pour créer un vocabulaire personnalisé

L'exemple `create-vocabulary` suivant crée un vocabulaire personnalisé. Pour créer un vocabulaire personnalisé, vous devez avoir créé un fichier texte contenant tous les termes que vous souhaitez transcrire de manière plus précise. Pour vocabulary-file-uri, spécifiez l'URI Amazon Simple Storage Service (Amazon S3) de ce fichier texte. Pour le code de langue, spécifiez un code de langue correspondant à la langue du vocabulaire personnalisé. Pour le nom de vocabulaire, spécifiez le nom que vous souhaitez donner au vocabulaire personnalisé.

```
aws transcribe create-vocabulary \
    --language-code language-code \
    --vocabulary-name cli-vocab-example \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-prefix/the-text-file-for-the-custom-vocabulary.txt
```

Sortie :

```
{
    "VocabularyName": "cli-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous CreateVocabularyà la section Référence des AWS CLI commandes.

`delete-language-model`

L'exemple de code suivant montre comment utiliser`delete-language-model`.

AWS CLI

Pour supprimer un modèle de langue personnalisé

L'`delete-language-model`exemple suivant supprime un modèle de langage personnalisé.

```
aws transcribe delete-language-model \
    --model-name model-name
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Amélioration de la précision de la transcription spécifique à un domaine grâce à des modèles de langue personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous DeleteLanguageModelà la section Référence des AWS CLI commandes.

`delete-medical-transcription-job`

L'exemple de code suivant montre comment utiliser`delete-medical-transcription-job`.

AWS CLI

Pour supprimer une tâche de transcription médicale

L'exemple `delete-medical-transcription-job` suivant crée une tâche de transcription médicale.

```
aws transcribe delete-medical-transcription-job \
    --medical-transcription-job-name medical-transcription-job-name
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteMedicalTranscriptionJoble guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous DeleteMedicalTranscriptionJobà la section Référence des AWS CLI commandes.

`delete-medical-vocabulary`

L'exemple de code suivant montre comment utiliser`delete-medical-vocabulary`.

AWS CLI

Pour supprimer un vocabulaire médical personnalisé

L'`delete-medical-vocabulary`exemple suivant supprime un vocabulaire médical personnalisé. Pour le nom du vocabulaire, spécifiez le nom du vocabulaire médical personnalisé.

```
aws transcribe delete-vocabulary \
    --vocabulary-name medical-custom-vocabulary-name
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Vocabulaires personnalisés médicaux dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous DeleteMedicalVocabularyà la section Référence des AWS CLI commandes.

`delete-transcription-job`

L'exemple de code suivant montre comment utiliser`delete-transcription-job`.

AWS CLI

Pour supprimer l'une des tâches de transcription

L'exemple `delete-transcription-job` suivant supprime l'une des tâches de transcription.

```
aws transcribe delete-transcription-job \
    --transcription-job-name your-transcription-job
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez DeleteTranscriptionJoble guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous DeleteTranscriptionJobà la section Référence des AWS CLI commandes.

`delete-vocabulary-filter`

L'exemple de code suivant montre comment utiliser`delete-vocabulary-filter`.

AWS CLI

Pour supprimer un filtre de vocabulaire

L'`delete-vocabulary-filter`exemple suivant supprime un filtre de vocabulaire.

```
aws transcribe delete-vocabulary-filter \
    --vocabulary-filter-name vocabulary-filter-name
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Filtrer les mots indésirables dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteVocabularyFilterà la section Référence des AWS CLI commandes.

`delete-vocabulary`

L'exemple de code suivant montre comment utiliser`delete-vocabulary`.

AWS CLI

Pour supprimer un vocabulaire personnalisé

L'exemple `delete-vocabulary` suivant supprime un vocabulaire personnalisé.

```
aws transcribe delete-vocabulary \
    --vocabulary-name vocabulary-name
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Vocabulaires personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous DeleteVocabularyà la section Référence des AWS CLI commandes.

`describe-language-model`

L'exemple de code suivant montre comment utiliser`describe-language-model`.

AWS CLI

Pour obtenir des informations sur un modèle de langage personnalisé spécifique

L'`describe-language-model`exemple suivant permet d'obtenir des informations sur un modèle de langage personnalisé spécifique. Par exemple, BaseModelName vous pouvez voir ci-dessous si votre modèle est entraîné à l'aide d'un WideBand modèle NarrowBand or. Les modèles de langage personnalisés dotés d'un modèle de NarrowBand base peuvent transcrire le son avec une fréquence d'échantillonnage inférieure à 16 kHz. Les modèles linguistiques utilisant un modèle WideBand de base peuvent transcrire le son avec une fréquence d'échantillonnage supérieure à 16 kHz. Le paramètre S3Uri indique le préfixe Amazon S3 que vous avez utilisé pour accéder aux données d'entraînement afin de créer le modèle de langage personnalisé.

```
aws transcribe describe-language-model \
    --model-name cli-clm-example
```

Sortie :

```
{
    "LanguageModel": {
        "ModelName": "cli-clm-example",
        "CreateTime": "2020-09-25T17:57:38.504000+00:00",
        "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",
        "LanguageCode": "language-code",
        "BaseModelName": "base-model-name",
        "ModelStatus": "IN_PROGRESS",
        "UpgradeAvailability": false,
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
            "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
            "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
        }
    }
```

```
}
```

Pour plus d'informations, consultez Amélioration de la précision de la transcription spécifique à un domaine grâce à des modèles de langue personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous DescribeLanguageModelà la section Référence des AWS CLI commandes.

`get-medical-transcription-job`

L'exemple de code suivant montre comment utiliser`get-medical-transcription-job`.

AWS CLI

Pour obtenir des informations sur une tâche de transcription médicale spécifique

L'`get-medical-transcription-job`exemple suivant permet d'obtenir des informations sur une tâche de transcription médicale spécifique. Pour accéder aux résultats de la transcription, utilisez le TranscriptFileUri paramètre. Si vous avez activé des fonctionnalités supplémentaires pour la tâche de transcription, vous pouvez les voir dans l'objet Paramètres. Le paramètre Spécialité indique la spécialité médicale du prestataire. Le paramètre Type indique si le discours utilisé dans la tâche de transcription est celui d'une conversation médicale ou d'une dictée médicale.

```
aws transcribe get-medical-transcription-job \
    --medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
        "TranscriptionJobStatus": "COMPLETED",
        "LanguageCode": "en-US",
        "MediaSampleRateHertz": 48000,
        "MediaFormat": "mp4",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-audio-file.file-extension"
```

```
        },
        "Transcript": {
            "TranscriptFileUri": "https://s3.Region.amazonaws.com/Amazon-S3-Prefix/
vocabulary-dictation-medical-transcription-job.json"
        },
        "StartTime": "2020-09-21T21:17:27.045000+00:00",
        "CreationTime": "2020-09-21T21:17:27.016000+00:00",
        "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
        "Settings": {
            "ChannelIdentification": false,
            "ShowAlternatives": false,
            "VocabularyName": "cli-medical-vocab-example"
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Pour plus d'informations, consultez Batch Transcription dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetMedicalTranscriptionJobà la section Référence des AWS CLI commandes.

`get-medical-vocabulary`

L'exemple de code suivant montre comment utiliser`get-medical-vocabulary`.

AWS CLI

Pour obtenir des informations sur un vocabulaire médical personnalisé

L'`get-medical-vocabulary`exemple suivant fournit des informations sur un vocabulaire médical personnalisé. Vous pouvez utiliser le VocabularyState paramètre pour voir l'état de traitement du vocabulaire. S'il est PRÊT, vous pouvez l'utiliser dans l' StartMedicalTranscriptionJob opération. :

```
aws transcribe get-medical-vocabulary \
    --vocabulary-name medical-vocab-example
```

Sortie :

```
{
    "VocabularyName": "medical-vocab-example",
    "LanguageCode": "en-US",
    "VocabularyState": "READY",
    "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
    "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-
medical-custom-vocabulary"
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés médicaux dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous GetMedicalVocabularyà la section Référence des AWS CLI commandes.

`get-transcription-job`

L'exemple de code suivant montre comment utiliser`get-transcription-job`.

AWS CLI

Pour obtenir des informations sur une tâche de transcription spécifique

L'exemple `get-transcription-job` suivant permet d'obtenir des informations sur une tâche de transcription spécifique. Pour accéder aux résultats de la transcription, utilisez le TranscriptFileUri paramètre. Utilisez le MediaFileUri paramètre pour voir quel fichier audio vous avez transcrit avec cette tâche. Vous pouvez utiliser l'objet Paramètres pour voir les fonctionnalités facultatives que vous avez activées dans la tâche de transcription.

```
aws transcribe get-transcription-job \
    --transcription-job-name your-transcription-job
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "your-transcription-job",
        "TranscriptionJobStatus": "COMPLETED",
        "LanguageCode": "language-code",
        "MediaSampleRateHertz": 48000,
        "MediaFormat": "mp4",
```

```
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-
extension"
        },
        "Transcript": {
            "TranscriptFileUri": "https://Amazon-S3-file-location-of-transcription-
output"
        },
        "StartTime": "2020-09-18T22:27:23.970000+00:00",
        "CreationTime": "2020-09-18T22:27:23.948000+00:00",
        "CompletionTime": "2020-09-18T22:28:21.197000+00:00",
        "Settings": {
            "ChannelIdentification": false,
            "ShowAlternatives": false
        },
        "IdentifyLanguage": true,
        "IdentifiedLanguageScore": 0.8672199249267578
    }
}
```

Pour plus d'informations, consultez Getting Started (interface de ligne de AWS commande) dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetTranscriptionJobà la section Référence des AWS CLI commandes.

`get-vocabulary-filter`

L'exemple de code suivant montre comment utiliser`get-vocabulary-filter`.

AWS CLI

Pour obtenir des informations sur un filtre de vocabulaire

L'`get-vocabulary-filter`exemple suivant permet d'obtenir des informations sur un filtre de vocabulaire. Vous pouvez utiliser le DownloadUri paramètre pour obtenir la liste des mots que vous avez utilisés pour créer le filtre de vocabulaire.

```
aws transcribe get-vocabulary-filter \
    --vocabulary-filter-name testFilter
```

Sortie :

```
{
    "VocabularyFilterName": "testFilter",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",
    "DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"
}
```

Pour plus d'informations, consultez la section Filtrer les mots indésirables dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetVocabularyFilterà la section Référence des AWS CLI commandes.

`get-vocabulary`

L'exemple de code suivant montre comment utiliser`get-vocabulary`.

AWS CLI

Pour obtenir des informations sur un vocabulaire personnalisé

L’exemple `get-vocabulary` suivant permet d’obtenir des informations sur un vocabulaire personnalisé précédemment créé.

```
aws transcribe get-vocabulary \
    --vocabulary-name cli-vocab-1
```

Sortie :

```
{
    "VocabularyName": "cli-vocab-1",
    "LanguageCode": "language-code",
    "VocabularyState": "READY",
    "LastModifiedTime": "2020-09-19T23:22:32.836000+00:00",
    "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-
custom-vocabulary"
}
```

Pour plus d’informations, consultez Vocabulaires personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous GetVocabularyà la section Référence des AWS CLI commandes.

`list-language-models`

L'exemple de code suivant montre comment utiliser`list-language-models`.

AWS CLI

Pour répertorier vos modèles linguistiques personnalisés

L'`list-language-models`exemple suivant répertorie les modèles linguistiques personnalisés associés à votre AWS compte et à votre région. Vous pouvez utiliser les `TuningDataS3Uri` paramètres `S3Uri` et pour trouver les préfixes Amazon S3 que vous avez utilisés comme données d'entraînement ou comme données de réglage. Vous BaseModelName indique si vous avez utilisé un NarrowBand ou un WideBand modèle pour créer un modèle de langage personnalisé. Vous pouvez transcrire du son avec une fréquence d'échantillonnage inférieure à 16 kHz à l'aide d'un modèle de langue personnalisé utilisant un modèle NarrowBand de base. Vous pouvez transcrire du son à 16 kHz ou plus avec un modèle de langage personnalisé à l'aide d'un modèle WideBand de base. Le `ModelStatus` paramètre indique si vous pouvez utiliser le modèle de langage personnalisé dans une tâche de transcription. Si la valeur est COMPLETED, vous pouvez l'utiliser dans un travail de transcription.

```
aws transcribe list-language-models
```

Sortie :

```
{
    "Models": [
        {
            "ModelName": "cli-clm-2",
            "CreateTime": "2020-09-25T17:57:38.504000+00:00",
            "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",
            "LanguageCode": "language-code",
            "BaseModelName": "WideBand",
            "ModelStatus": "IN_PROGRESS",
            "UpgradeAvailability": false,
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/clm-training-data/",
                "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/clm-tuning-data/",
```

```
                "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
used-to-create-the-custom-language-model"
            }
        },
        {
            "ModelName": "cli-clm-1",
            "CreateTime": "2020-09-25T17:16:01.835000+00:00",
            "LastModifiedTime": "2020-09-25T17:16:15.555000+00:00",
            "LanguageCode": "language-code",
            "BaseModelName": "WideBand",
            "ModelStatus": "IN_PROGRESS",
            "UpgradeAvailability": false,
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/clm-training-data/",
                "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
used-to-create-the-custom-language-model"
            }
        },
        {
            "ModelName": "clm-console-1",
            "CreateTime": "2020-09-24T19:26:28.076000+00:00",
            "LastModifiedTime": "2020-09-25T04:25:22.271000+00:00",
            "LanguageCode": "language-code",
            "BaseModelName": "NarrowBand",
            "ModelStatus": "COMPLETED",
            "UpgradeAvailability": false,
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/clm-training-data/",
                "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
used-to-create-the-custom-language-model"
            }
        }
    ]
}
```

Pour plus d'informations, consultez Amélioration de la précision de la transcription spécifique à un domaine grâce à des modèles de langue personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous ListLanguageModelsà la section Référence des AWS CLI commandes.

`list-medical-transcription-jobs`

L'exemple de code suivant montre comment utiliser`list-medical-transcription-jobs`.

AWS CLI

Pour répertorier les tâches de transcription médicale

L'`list-medical-transcription-jobs`exemple suivant répertorie les tâches de transcription médicale associées à votre AWS compte et à votre région. Pour obtenir plus d'informations sur une tâche de transcription particulière, copiez la valeur d'un MedicalTranscriptionJobName paramètre dans le résultat de transcription et spécifiez cette valeur pour l'`MedicalTranscriptionJobName`option de la `get-medical-transcription-job` commande. Pour voir d'autres tâches de transcription, copiez la valeur du NextToken paramètre, réexécutez la `list-medical-transcription-jobs` commande et spécifiez cette valeur dans l'`--next-token`option.

```
aws transcribe list-medical-transcription-jobs
```

Sortie :

```
{
    "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn44ON/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z1O2FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqlZB10Wa9McC+ebOl
+LaDtZPC4u6ttoHLRlEfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYAl5VTDmXXzQPBQOHPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBsj75+2vQZOSVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6JgOcdVPtEg8PlDr5kgYYG3q3OmYXX37U3FZuJmnTI63VtIXsNnOU5eGoYObtpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eUOOD8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYoX3tTPWlD7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GAOWiBpfvAZRvEb924i6yGHyMC7y54O1ZAwSBupmI
+FFd13CaPO4kN1vJlth6aM5vUPXg4BpyUhtbRhwD/KxCvf9K0tLJGyL1A==",
    "MedicalTranscriptionJobSummaries": [
        {
            "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
```

```
            "CreationTime": "2020-09-21T21:17:27.016000+00:00",
            "StartTime": "2020-09-21T21:17:27.045000+00:00",
            "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "DICTATION"
        },
        {
            "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
            "CreationTime": "2020-09-21T21:01:14.569000+00:00",
            "StartTime": "2020-09-21T21:01:14.592000+00:00",
            "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "DICTATION"
        },
        {
            "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
            "CreationTime": "2020-09-21T19:09:18.171000+00:00",
            "StartTime": "2020-09-21T19:09:18.199000+00:00",
            "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "CONVERSATION"
        },
        {
            "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
            "CreationTime": "2020-09-21T18:43:37.157000+00:00",
            "StartTime": "2020-09-21T18:43:37.265000+00:00",
            "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "CONVERSATION"
```

```
        },
        {
            "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
            "CreationTime": "2020-09-20T23:46:44.053000+00:00",
            "StartTime": "2020-09-20T23:46:44.081000+00:00",
            "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "CONVERSATION"
        }
    ]
}
```

Pour plus d'informations, consultez https://docs.aws.amazon.com/transcribe/ latest/dg/batch - med-transcription.html> dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListMedicalTranscriptionJobsà la section Référence des AWS CLI commandes.

`list-medical-vocabularies`

L'exemple de code suivant montre comment utiliser`list-medical-vocabularies`.

AWS CLI

Pour répertorier vos vocabulaires médicaux personnalisés

L'`list-medical-vocabularies`exemple suivant répertorie les vocabulaires médicaux personnalisés associés à votre AWS compte et à votre région. Pour obtenir plus d'informations sur une tâche de transcription particulière, copiez la valeur d'un `MedicalTranscriptionJobName` paramètre dans le résultat de transcription et spécifiez cette valeur pour l'`MedicalTranscriptionJobName`option de la `get-medical-transcription-job` commande. Pour voir d'autres tâches de transcription, copiez la valeur du `NextToken` paramètre, réexécutez la `list-medical-transcription-jobs` commande et spécifiez cette valeur dans l'`--next-token`option.

```
aws transcribe list-medical-vocabularies
```

Sortie :

```
{
    "Vocabularies": [
        {
            "VocabularyName": "cli-medical-vocab-2",
            "LanguageCode": "en-US",
            "LastModifiedTime": "2020-09-21T21:44:59.521000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "cli-medical-vocab-1",
            "LanguageCode": "en-US",
            "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
            "VocabularyState": "READY"
        }
    ]
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés médicaux dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous ListMedicalVocabulariesà la section Référence des AWS CLI commandes.

`list-transcription-jobs`

L'exemple de code suivant montre comment utiliser`list-transcription-jobs`.

AWS CLI

Pour répertorier les tâches de transcription

L'`list-transcription-jobs`exemple suivant répertorie les tâches de transcription associées à votre AWS compte et à votre région.

```
aws transcribe list-transcription-jobs
```

Sortie :

```
{
    "NextToken": "NextToken",
```

```
    "TranscriptionJobSummaries": [
        {
            "TranscriptionJobName": "speak-id-job-1",
            "CreationTime": "2020-08-17T21:06:15.391000+00:00",
            "StartTime": "2020-08-17T21:06:15.416000+00:00",
            "CompletionTime": "2020-08-17T21:07:05.098000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "job-1",
            "CreationTime": "2020-08-17T20:50:24.207000+00:00",
            "StartTime": "2020-08-17T20:50:24.230000+00:00",
            "CompletionTime": "2020-08-17T20:52:18.737000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "sdk-test-job-4",
            "CreationTime": "2020-08-17T20:32:27.917000+00:00",
            "StartTime": "2020-08-17T20:32:27.956000+00:00",
            "CompletionTime": "2020-08-17T20:33:15.126000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "Diarization-speak-id",
            "CreationTime": "2020-08-10T22:10:09.066000+00:00",
            "StartTime": "2020-08-10T22:10:09.116000+00:00",
            "CompletionTime": "2020-08-10T22:26:48.172000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "your-transcription-job-name",
            "CreationTime": "2020-07-29T17:45:09.791000+00:00",
            "StartTime": "2020-07-29T17:45:09.826000+00:00",
            "CompletionTime": "2020-07-29T17:46:20.831000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
```

```
            "OutputLocationType": "SERVICE_BUCKET"
        }
    ]
}
```

Pour plus d'informations, consultez Getting Started (interface de ligne de AWS commande) dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListTranscriptionJobsà la section Référence des AWS CLI commandes.

`list-vocabularies`

L'exemple de code suivant montre comment utiliser`list-vocabularies`.

AWS CLI

Pour répertorier les vocabulaires personnalisés

L'`list-vocabularies`exemple suivant répertorie les vocabulaires personnalisés associés à votre AWS compte et à votre région.

```
aws transcribe list-vocabularies
```

Sortie :

```
{
    "NextToken": "NextToken",
    "Vocabularies": [
        {
            "VocabularyName": "ards-test-1",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-27T22:00:27.330000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "sample-test",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T23:04:11.044000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "CRLF-to-LF-test-3-1",
```

```
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T22:12:22.277000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "CRLF-to-LF-test-2",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T21:53:50.455000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "CRLF-to-LF-1-1",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T21:39:33.356000+00:00",
            "VocabularyState": "READY"
        }
    ]
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous ListVocabulariesà la section Référence des AWS CLI commandes.

`list-vocabulary-filters`

L'exemple de code suivant montre comment utiliser`list-vocabulary-filters`.

AWS CLI

Pour répertorier vos filtres de vocabulaire

L'`list-vocabulary-filters`exemple suivant répertorie les filtres de vocabulaire associés à votre AWS compte et à votre région.

```
aws transcribe list-vocabulary-filters
```

Sortie :

```
{
    "NextToken": "NextToken": [
        {
```

```
            "VocabularyFilterName": "testFilter",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00"
        },
        {
            "VocabularyFilterName": "testFilter2",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-21T23:29:35.174000+00:00"
        },
        {
            "VocabularyFilterName": "filter2",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-08T20:18:26.426000+00:00"
        },
        {
            "VocabularyFilterName": "filter-review",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-06-03T18:52:30.448000+00:00"
        },
        {
            "VocabularyFilterName": "crlf-filt",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-22T19:42:42.737000+00:00"
        }
    ]
}
```

Pour plus d'informations, consultez la section Filtrer les mots indésirables dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListVocabularyFilterssà la section Référence des AWS CLI commandes.

`start-medical-transcription-job`

L'exemple de code suivant montre comment utiliser`start-medical-transcription-job`.

AWS CLI

Exemple 1 : Transcrire une dictée médicale enregistrée sous forme de fichier audio

L’exemple `start-medical-transcription-job` suivant transcrit un fichier audio. Vous spécifiez l’emplacement de la sortie de transcription dans le paramètre `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfile.json
```

Contenu de `myfile.json` :

```
{
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T00:35:22.256000+00:00",
        "CreationTime": "2020-09-20T00:35:22.218000+00:00",
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Pour plus d'informations, consultez Présentation de la transcription par lots dans le Guide du développeur Amazon Transcribe.

Exemple 2 : Transcrire un dialogue entre un médecin et son patient enregistré sous forme de fichier audio

L'exemple `start-medical-transcription-job` suivant transcrit un fichier audio contenant un dialogue entre un médecin et son patient. Vous spécifiez l'emplacement de la sortie de transcription dans le OutputBucketName paramètre.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysecondfile.json
```

Contenu de `mysecondfile.json` :

```
{
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-conversation-medical-transcription-
job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T23:19:49.965000+00:00",
        "CreationTime": "2020-09-20T23:19:49.941000+00:00",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

Pour plus d'informations, consultez Présentation de la transcription par lots dans le Guide du développeur Amazon Transcribe.

Exemple 3 : Transcrire un fichier audio multicanal d'un dialogue entre un médecin et son patient

L'exemple `start-medical-transcription-job` suivant transcrit le son de chaque canal du fichier audio et fusionne les transcriptions distinctes de chaque canal en une seule sortie de transcription. Vous spécifiez l'emplacement de la sortie de transcription dans le paramètre `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mythirdfile.json
```

Contenu de `mythirdfile.json` :

```
{
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
        "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "Settings":{
          "ChannelIdentification": true
        }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T23:46:44.081000+00:00",
        "CreationTime": "2020-09-20T23:46:44.053000+00:00",
        "Settings": {
            "ChannelIdentification": true
        },
        "Specialty": "PRIMARYCARE",
```

```
        "Type": "CONVERSATION"
    }
}
```

Pour plus d'informations, consultez Identification des canaux dans le Guide du développeur Amazon Transcribe.

Exemple 4 : Transcrire un fichier audio d'un dialogue entre un médecin et son patient et identifier les locuteurs dans la sortie de transcription

L'exemple `start-medical-transcription-job` suivant transcrit le fichier audio et étiquette le discours de chaque locuteur dans la sortie de transcription. Vous spécifiez l'emplacement de la sortie de transcription dans le paramètre `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfourthfile.json
```

Contenu de `myfourthfile.json` :

```
{
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
    "Settings":{
        "ShowSpeakerLabels": true,
        "MaxSpeakerLabels": 2
        }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
```

```
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T18:43:37.265000+00:00",
        "CreationTime": "2020-09-21T18:43:37.157000+00:00",
        "Settings": {
            "ShowSpeakerLabels": true,
            "MaxSpeakerLabels": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

Pour plus d'informations, consultez Identification des locuteurs dans le Guide du développeur Amazon Transcribe.

Exemple 5 : Transcrire une conversation médicale enregistrée sous forme de fichier audio avec jusqu'à deux alternatives de transcription

L'exemple `start-medical-transcription-job` suivant crée jusqu'à deux transcriptions alternatives à partir d'un seul fichier audio. Un niveau de confiance est associé à chaque transcription. Par défaut, Amazon Transcribe renvoie la transcription avec le niveau de confiance le plus élevé. Vous pouvez spécifier qu'Amazon Transcribe renvoie des transcriptions supplémentaires avec des niveaux de confiance inférieurs. Vous spécifiez l'emplacement de la sortie de transcription dans le paramètre `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfifthfile.json
```

Contenu de `myfifthfile.json` :

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
```

```
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
        "ShowAlternatives": true,
        "MaxAlternatives": 2
    }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T19:09:18.199000+00:00",
        "CreationTime": "2020-09-21T19:09:18.171000+00:00",
        "Settings": {
            "ShowAlternatives": true,
            "MaxAlternatives": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

Pour plus d'informations, consultez Transcriptions alternatives dans le Guide du développeur Amazon Transcribe.

Exemple 6 : Transcrire le fichier audio d'une dictée médicale avec jusqu'à deux transcriptions alternatives

L'exemple `start-medical-transcription-job` suivant transcrit un fichier audio et utilise un filtre de vocabulaire pour masquer les mots indésirables. Vous spécifiez l'emplacement de la sortie de transcription dans le OutputBucketName paramètre.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysixthfile.json
```

Contenu de `mysixthfile.json` :

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
          "ShowAlternatives": true,
          "MaxAlternatives": 2
    }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T21:01:14.592000+00:00",
        "CreationTime": "2020-09-21T21:01:14.569000+00:00",
        "Settings": {
            "ShowAlternatives": true,
            "MaxAlternatives": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Pour plus d'informations, consultez Transcriptions alternatives dans le Guide du développeur Amazon Transcribe.

Exemple 7 : Transcrire un fichier audio d'une dictée médicale avec une précision accrue en utilisant un vocabulaire personnalisé

L'exemple `start-medical-transcription-job` suivant transcrit un fichier audio et utilise un vocabulaire médical personnalisé que vous avez créé précédemment pour augmenter la précision de la transcription. Vous spécifiez l'emplacement de la sortie de transcription dans le paramètre `OutputBucketName`.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

Contenu de `mysixthfile.json` :

```
{
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
        "VocabularyName": "cli-medical-vocab-1"
    }
}
```

Sortie :

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T21:17:27.045000+00:00",
        "CreationTime": "2020-09-21T21:17:27.016000+00:00",
        "Settings": {
            "VocabularyName": "cli-medical-vocab-1"
```

```
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés médicaux dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous StartMedicalTranscriptionJobà la section Référence des AWS CLI commandes.

`start-transcription-job`

L'exemple de code suivant montre comment utiliser`start-transcription-job`.

AWS CLI

Exemple 1 : Transcrire un fichier audio

L'exemple `start-transcription-job` suivant transcrit un fichier audio.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfile.json
```

Contenu de `myfile.json` :

```
{
    "TranscriptionJobName": "cli-simple-transcription-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    }
}
```

Pour plus d'informations, consultez Getting Started (interface de ligne de AWS commande) dans le manuel Amazon Transcribe Developer Guide.

Exemple 2 : Transcrire un fichier audio multicanal

L'exemple `start-transcription-job` suivant transcrit le fichier audio multicanal.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysecondfile.json
```

Contenu de `mysecondfile.json` :

```
{
    "TranscriptionJobName": "cli-channelid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "ChannelIdentification":true
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-channelid-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-17T16:07:56.817000+00:00",
        "CreationTime": "2020-09-17T16:07:56.784000+00:00",
        "Settings": {
            "ChannelIdentification": true
        }
    }
}
```

Pour plus d'informations, consultez Transcription de l'audio multicanal dans le Guide du développeur Amazon Transcribe.

Exemple 3 : Transcrire un fichier audio et identifier les différents locuteurs

L'exemple `start-transcription-job` suivant transcrit le fichier audio et identifie les locuteurs dans la sortie de transcription.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mythirdfile.json
```

Contenu de `mythirdfile.json` :

```
{
    "TranscriptionJobName": "cli-speakerid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-speakerid-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-17T16:22:59.696000+00:00",
        "CreationTime": "2020-09-17T16:22:59.676000+00:00",
        "Settings": {
            "ShowSpeakerLabels": true,
            "MaxSpeakerLabels": 2
        }
    }
}
```

Pour plus d'informations, consultez Identification des locuteurs dans le Guide du développeur Amazon Transcribe.

Exemple 4 : Transcrire un fichier audio et masquer les mots indésirables dans la sortie de transcription

L'exemple `start-transcription-job` suivant transcrit un fichier audio et utilise un filtre de vocabulaire que vous avez créé précédemment pour masquer les mots indésirables.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfourthfile.json
```

Contenu de `myfourthfile.json` :

```
{
    "TranscriptionJobName": "cli-filter-mask-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "mask"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-filter-mask-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyFilterName": "your-vocabulary-filter",
            "VocabularyFilterMethod": "mask"
```

```
        }
    }
}
```

Pour plus d'informations, consultez Filtrage des transcriptions dans le Guide du développeur Amazon Transcribe.

Exemple 5 : Transcrire un fichier audio et supprimer les mots indésirables dans la sortie de transcription

L'exemple `start-transcription-job` suivant transcrit un fichier audio et utilise un filtre de vocabulaire que vous avez créé précédemment pour masquer les mots indésirables.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfifthfile.json
```

Contenu de `myfifthfile.json` :

```
{
    "TranscriptionJobName": "cli-filter-remove-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "remove"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-filter-remove-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
```

```
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyFilterName": "your-vocabulary-filter",
            "VocabularyFilterMethod": "remove"
        }
    }
}
```

Pour plus d'informations, consultez Filtrage des transcriptions dans le Guide du développeur Amazon Transcribe.

Exemple 6 : Transcrire un fichier audio avec une précision accrue en utilisant un vocabulaire personnalisé

L'exemple `start-transcription-job` suivant transcrit un fichier audio et utilise un filtre de vocabulaire que vous avez créé précédemment pour masquer les mots indésirables.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysixthfile.json
```

Contenu de `mysixthfile.json` :

```
{
    "TranscriptionJobName": "cli-vocab-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "VocabularyName": "your-vocabulary"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-vocab-job",
```

```
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyName": "your-vocabulary"
        }
    }
}
```

Pour plus d'informations, consultez Filtrage des transcriptions dans le Guide du développeur Amazon Transcribe.

Exemple 7 : Identifier la langue d'un fichier audio et le transcrire

L'exemple `start-transcription-job` suivant transcrit un fichier audio et utilise un filtre de vocabulaire que vous avez créé précédemment pour masquer les mots indésirables.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

Contenu de `myseventhfile.json` :

```
{
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "IdentifyLanguage": true,
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-identify-language-transcription-job",
```

```
        "TranscriptionJobStatus": "IN_PROGRESS",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-18T22:27:23.970000+00:00",
        "CreationTime": "2020-09-18T22:27:23.948000+00:00",
        "IdentifyLanguage": true
    }
}
```

Pour plus d'informations, consultez Identification de la langue dans le Guide du développeur Amazon Transcribe.

Exemple 8 : Transcrire un fichier audio contenant des données d'identification personnelle expurgées

L'exemple `start-transcription-job` suivant transcrit le fichier audio et expurge les données d'identification personnelle dans la sortie de transcription.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myeighthfile.json
```

Contenu de `myeigthfile.json` :

```
{
    "TranscriptionJobName": "cli-redaction-job",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "ContentRedaction": {
        "RedactionOutput":"redacted",
        "RedactionType":"PII"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
```

```
        "TranscriptionJobName": "cli-redaction-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
        "StartTime": "2020-09-25T23:49:13.195000+00:00",
        "CreationTime": "2020-09-25T23:49:13.176000+00:00",
        "ContentRedaction": {
            "RedactionType": "PII",
            "RedactionOutput": "redacted"
        }
    }
}
```

Pour plus d'informations, consultez Expurgation automatique du contenu dans le Guide du développeur Amazon Transcribe.

Exemple 9 : Générer une transcription contenant des données d'identification personnelle (PII) expurgées et une transcription non expurgée

L'exemple `start-transcription-job` suivant génère deux transcriptions du fichier audio, l'une avec les données d'identification personnelle expurgées, et l'autre sans aucune expurgation.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myninthfile.json
```

Contenu de `myninthfile.json` :

```
{
    "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",
    "LanguageCode": "language-code",
    "Media": {
          "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
    "ContentRedaction": {
        "RedactionOutput":"redacted_and_unredacted",
        "RedactionType":"PII"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
        "StartTime": "2020-09-25T23:59:47.677000+00:00",
        "CreationTime": "2020-09-25T23:59:47.653000+00:00",
        "ContentRedaction": {
            "RedactionType": "PII",
            "RedactionOutput": "redacted_and_unredacted"
        }
    }
}
```

Pour plus d'informations, consultez Expurgation automatique du contenu dans le Guide du développeur Amazon Transcribe.

Exemple 10 : Utiliser un modèle de langage personnalisé que vous avez créé précédemment pour transcrire un fichier audio.

L'exemple `start-transcription-job` suivant transcrit le fichier audio avec un modèle de langage personnalisé que vous avez créé précédemment.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mytenthfile.json
```

Contenu de `mytenthfile.json` :

```
{
    "TranscriptionJobName": "cli-clm-2-job-1",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-extension"
    },
    "ModelSettings": {
        "LanguageModelName":"cli-clm-2"
    }
}
```

Sortie :

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-clm-2-job-1",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-
extension"
        },
        "StartTime": "2020-09-28T17:56:01.835000+00:00",
        "CreationTime": "2020-09-28T17:56:01.801000+00:00",
        "ModelSettings": {
            "LanguageModelName": "cli-clm-2"
        }
    }
}
```

Pour plus d'informations, consultez Amélioration de la précision de la transcription spécifique à un domaine grâce à des modèles de langue personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous StartTranscriptionJobà la section Référence des AWS CLI commandes.

`update-medical-vocabulary`

L'exemple de code suivant montre comment utiliser`update-medical-vocabulary`.

AWS CLI

Mettre à jour un vocabulaire médical personnalisé avec de nouveaux termes.

L'`update-medical-vocabulary`exemple suivant remplace les termes utilisés dans un vocabulaire médical personnalisé par les nouveaux termes. Prérequis : pour remplacer les termes d'un vocabulaire médical personnalisé, vous avez besoin d'un fichier contenant de nouveaux termes.

```
aws transcribe update-medical-vocabulary \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/medical-custom-
vocabulary.txt \
```

```
    --vocabulary-name medical-custom-vocabulary \
    --language-code language
```

Sortie :

```
{
    "VocabularyName": "medical-custom-vocabulary",
    "LanguageCode": "en-US",
    "VocabularyState": "PENDING"
}
```

Pour plus d'informations, consultez Vocabulaires personnalisés médicaux dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous UpdateMedicalVocabularyà la section Référence des AWS CLI commandes.

`update-vocabulary-filter`

L'exemple de code suivant montre comment utiliser`update-vocabulary-filter`.

AWS CLI

Pour remplacer les mots dans un filtre de vocabulaire

L'`update-vocabulary-filter`exemple suivant remplace les mots d'un filtre de vocabulaire par de nouveaux mots. Prérequis : pour mettre à jour un filtre de vocabulaire avec les nouveaux mots, vous devez enregistrer ces mots sous forme de fichier texte.

```
aws transcribe update-vocabulary-filter \
    --vocabulary-filter-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/your-text-file-to-update-your-vocabulary-filter.txt \
    --vocabulary-filter-name vocabulary-filter-name
```

Sortie :

```
{
    "VocabularyFilterName": "vocabulary-filter-name",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"
```

```
}
```

Pour plus d'informations, consultez la section Filtrer les mots indésirables dans le manuel Amazon Transcribe Developer Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateVocabularyFilterà la section Référence des AWS CLI commandes.

`update-vocabulary`

L'exemple de code suivant montre comment utiliser`update-vocabulary`.

AWS CLI

Pour mettre à jour un vocabulaire personnalisé avec de nouveaux termes.

L’exemple `update-vocabulary` suivant remplace les termes utilisés pour créer un vocabulaire personnalisé par les nouveaux termes que vous fournissez. Prérequis : pour remplacer les termes d’un vocabulaire personnalisé, vous avez besoin d’un fichier contenant les nouveaux termes.

```
aws transcribe update-vocabulary \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/custom-
vocabulary.txt \
    --vocabulary-name custom-vocabulary \
    --language-code language-code
```

Sortie :

```
{
    "VocabularyName": "custom-vocabulary",
    "LanguageCode": "language",
    "VocabularyState": "PENDING"
}
```

Pour plus d’informations, consultez Vocabulaires personnalisés dans le Guide du développeur Amazon Transcribe.

- Pour plus de détails sur l'API, reportez-vous UpdateVocabularyà la section Référence des AWS CLI commandes.

Exemples d'Amazon Translate utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon Translate.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`import-terminology`

L'exemple de code suivant montre comment utiliser`import-terminology`.

AWS CLI

Pour importer une terminologie personnalisée à partir d'un fichier

L'`import-terminology`exemple suivant crée une terminologie appelée `MyTestTerminology` à partir du `test-terminology.csv` fichier :

```
aws translate import-terminology \
    --name MyTestTerminology \
    --description "Creating a test terminology in AWS Translate" \
    --merge-strategy OVERWRITE \
    --data-file fileb://test-terminology.csv \
    --terminology-data Format=CSV
```

Contenu de `test-terminology.csv` :

en, fr, es, zh Bonjour tout le monde ! , Bonjour tout le monde ! , Bonjour le monde ! , ? ? ? ? Amazon, Amazon, Amazon, Amazon

Sortie :

```
{
    "TerminologyProperties": {
        "SourceLanguageCode": "en",
        "Name": "MyTestTerminology",
        "TargetLanguageCodes": [
            "fr",
            "es",
            "zh"
        ],
        "SizeBytes": 97,
        "LastUpdatedAt": 1571089500.851,
        "CreatedAt": 1571089500.851,
        "TermCount": 6,
        "Arn": "arn:aws:translate:us-west-2:123456789012:terminology/
MyTestTerminology/LATEST",
        "Description": "Creating a test terminology in AWS Translate"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ImportTerminologyà la section Référence des AWS CLI commandes.

Trusted Advisor exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with Trusted Advisor.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-organization-recommendation`

L'exemple de code suivant montre comment utiliser`get-organization-recommendation`.

AWS CLI

Pour obtenir une recommandation d'organisation

L'`get-organization-recommendation`exemple suivant obtient une recommandation d'organisation par son identifiant.

```
aws trustedadvisor get-organization-recommendation \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

Sortie :

```
{
    "organizationRecommendation": {
        "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
        "name": "Lambda Runtime Deprecation Warning",
        "description": "One or more lambdas are using a deprecated runtime",
        "awsServices": [
            "lambda"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
        "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
        "lifecycleStage": "resolved",
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "warning",
        "type": "priority"
    }
```

```
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous GetOrganizationRecommendationà la section Référence des AWS CLI commandes.

`get-recommendation`

L'exemple de code suivant montre comment utiliser`get-recommendation`.

AWS CLI

Pour obtenir une recommandation

L'`get-recommendation`exemple suivant obtient une recommandation par son identifiant.

```
aws trustedadvisor get-recommendation \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578
```

Sortie :

```
{
    "recommendation": {
        "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
        "name": "MFA Recommendation",
        "description": "Enable multi-factor authentication",
        "awsServices": [
            "iam"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
        "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
        "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
```

```
        },
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 1,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "error",
        "type": "standard"
    }
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous GetRecommendationà la section Référence des AWS CLI commandes.

`list-checks`

L'exemple de code suivant montre comment utiliser`list-checks`.

AWS CLI

Pour répertorier les chèques Trusted Advisor

L'`list-checks`exemple suivant répertorie tous les contrôles Trusted Advisor.

```
aws trustedadvisor list-checks
```

Sortie :

```
{
    "checkSummaries": [
        {
            "arn": "arn:aws:trustedadvisor:::check/1iG5NDGVre",
            "awsServices": [
                "EC2"
            ],
```

```
            "description": "Checks security groups for rules that allow unrestricted
access to a resource. Unrestricted access increases opportunities for malicious
activity (hacking, denial-of-service attacks, loss of data)",
            "id": "1iG5NDGVre",
            "metadata": {
                "0": "Region",
                "1": "Security Group Name",
                "2": "Security Group ID",
                "3": "Protocol",
                "4": "Port",
                "5": "Status",
                "6": "IP Range"
            },
            "name": "Security Groups - Unrestricted Access",
            "pillars": [
                "security"
            ],
            "source": "ta_check"
        },
        {
            "arn": "arn:aws:trustedadvisor:::check/1qazXsw23e",
            "awsServices": [
                "RDS"
            ],
            "description": "Checks your usage of RDS and provides recommendations
on purchase of Reserved Instances to help reduce costs incurred from using RDS
On-Demand. AWS generates these recommendations by analyzing your On-Demand usage
for the past 30 days. We then simulate every combination of reservations in the
generated category of usage in order to identify the best number of each type
of Reserved Instance to purchase to maximize your savings. This check covers
recommendations based on partial upfront payment option with 1-year or 3-year
commitment. This check is not available to accounts linked in Consolidated Billing.
Recommendations are only available for the Paying Account.",
            "id": "1qazXsw23e",
            "metadata": {
                "0": "Region",
                "1": "Family",
                "2": "Instance Type",
                "3": "License Model",
                "4": "Database Edition",
                "5": "Database Engine",
                "6": "Deployment Option",
                "7": "Recommended number of Reserved Instances to purchase",
                "8": "Expected Average Reserved Instance Utilization",
```

```
                "9": "Estimated Savings with Recommendation (monthly)"
                "10": "Upfront Cost of Reserved Instances",
                "11": "Estimated cost of Reserved Instances (monthly)",
                "12": "Estimated On-Demand Cost Post Recommended Reserved Instance
Purchase (monthly)",
                "13": "Estimated Break Even (months)",
                "14": "Lookback Period (days)",
                "15": "Term (years)"
            },
            "name": "Amazon Relational Database Service (RDS) Reserved Instance
Optimization",
            "pillars": [
                "cost_optimizing"
            ],
            "source": "ta_check"
        },
        {
            "arn": "arn:aws:trustedadvisor:::check/1qw23er45t",
            "awsServices": [
                "Redshift"
            ],
            "description": "Checks your usage of Redshift and provides
recommendations on purchase of Reserved Nodes to help reduce costs incurred from
using Redshift On-Demand. AWS generates these recommendations by analyzing your
On-Demand usage for the past 30 days. We then simulate every combination of
reservations in the generated category of usage in order to identify the best
number of each type of Reserved Nodes to purchase to maximize your savings. This
check covers recommendations based on partial upfront payment option with 1-year or
3-year commitment. This check is not available to accounts linked in Consolidated
Billing. Recommendations are only available for the Paying Account.",
            "id": "1qw23er45t",
            "metadata": {
                "0": "Region",
                "1": "Family",
                "2": "Node Type",
                "3": "Recommended number of Reserved Nodes to purchase",
                "4": "Expected Average Reserved Node Utilization",
                "5": "Estimated Savings with Recommendation (monthly)",
                "6": "Upfront Cost of Reserved Nodes",
                "7": "Estimated cost of Reserved Nodes (monthly)",
                "8": "Estimated On-Demand Cost Post Recommended Reserved Nodes
Purchase (monthly)",
                "9": "Estimated Break Even (months)",
                "10": "Lookback Period (days)",
```

```
                "11": "Term (years)",
            },
            "name": "Amazon Redshift Reserved Node Optimization",
            "pillars": [
                "cost_optimizing"
            ],
            "source": "ta_check"
        },
    ],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous ListChecksà la section Référence des AWS CLI commandes.

`list-organization-recommendation-accounts`

L'exemple de code suivant montre comment utiliser`list-organization-recommendation-accounts`.

AWS CLI

Pour répertorier les comptes de recommandation d'une organisation

L'`list-organization-recommendation-accounts`exemple suivant répertorie tous les résumés des recommandations de compte pour une recommandation d'organisation par identifiant.

```
aws trustedadvisor list-organization-recommendation-accounts \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

Sortie :

```
{
    "accountRecommendationLifecycleSummaries": [{
        "accountId": "000000000000",
```

```
        "accountRecommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
        "lifecycleStage": "resolved",
        "updateReason": "Resolved issue",
        "updateReasonCode": "valid_business_case",
        "lastUpdatedAt": "2023-01-17T18:25:44.552Z"
    }],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous ListOrganizationRecommendationAccountsà la section Référence des AWS CLI commandes.

`list-organization-recommendation-resources`

L'exemple de code suivant montre comment utiliser`list-organization-recommendation-resources`.

AWS CLI

Pour répertorier les ressources de recommandation d'une organisation

L'`list-organization-recommendation-resources`exemple suivant répertorie toutes les ressources pour une recommandation d'organisation en fonction de son identifiant.

```
aws trustedadvisor list-organization-recommendation-resources \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

Sortie :

```
{
    "organizationRecommendationResourceSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
bb38affc0ce0681d9a6cd13f30238ba03a8f63dfe7a379dc403c619119d86af",
            "awsResourceId": "database-1-instance-1",
```

```
            "id":
 "bb38affc0ce0681d9a6cd13f302383ba03a8f63dfe7a379dc403c619119d86af",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "208.79999999999998",
                "2": "database-1-instance-1",
                "3": "db.r5.large",
                "4": "false",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-1-instance-1",
                "7": "1"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-
ae72-730598fa89d0/51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
            "awsResourceId": "database-1",
            "id":
 "51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "31.679999999999996",
                "2": "database-1",
                "3": "db.t3.small",
                "4": "false",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-1",
                "7": "20"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
        {
```

```
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
            "awsResourceId": "database-2-instance-1-us-west-2a",
            "id":
 "f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "187.20000000000002",
                "2": "database-2-instance-1-us-west-2a",
                "3": "db.r6g.large",
                "4": "true",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-2-instance-1-
us-west-2a",
                "7": "1"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
    ],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous ListOrganizationRecommendationResourcesà la section Référence des AWS CLI commandes.

`list-organization-recommendations`

L'exemple de code suivant montre comment utiliser`list-organization-recommendations`.

AWS CLI

Exemple 1 : Pour répertorier les recommandations d'une organisation

L'`list-organization-recommendations`exemple suivant répertorie toutes les recommandations relatives à l'organisation et n'inclut aucun filtre.

```
aws trustedadvisor list-organization-recommendations
```

Sortie :

```
{
    "organizationRecommendationSummaries": [
        {
            "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
            "name": "Lambda Runtime Deprecation Warning",
            "awsServices": [
                "lambda"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
            "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
            "lifecycleStage": "resolved",
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 0,
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "warning",
            "type": "priority"
        },
        {
            "arn": "arn:aws:trustedadvisor:::organization-
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
            "name": "Lambda Runtime Deprecation Warning",
            "awsServices": [
                "lambda"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
            "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
            "lifecycleStage": "resolved",
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 0,
```

```
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "warning",
            "type": "priority"
        },
    ],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

Exemple 2 : pour répertorier les recommandations d'une organisation à l'aide d'un filtre

L'`list-organization-recommendations`exemple suivant filtre et renvoie au maximum une recommandation d'organisation faisant partie du pilier « sécurité ».

```
aws trustedadvisor list-organization-recommendations \
    --pillar security \
    --max-items 100
```

Sortie :

```
{
    "organizationRecommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
        "name": "Lambda Runtime Deprecation Warning",
        "awsServices": [
            "lambda"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
        "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
        "lifecycleStage": "resolved",
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 0,
            "warningCount": 0
```

```
        },
        "source": "ta_check",
        "status": "warning",
        "type": "priority"
    }],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

Exemple 3 : pour répertorier les recommandations d'une organisation à l'aide d'un jeton de pagination

L'`list-organization-recommendations`exemple suivant utilise le « NextToken » renvoyé par une demande précédente pour récupérer la page suivante des recommandations de l'organisation.

```
aws trustedadvisor list-organization-recommendations \
    --pillar security \
    --max-items 100 \
    --starting-token <next-token>
```

Sortie :

```
{
    "organizationRecommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor:::organization-
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
        "name": "Lambda Runtime Deprecation Warning",
        "awsServices": [
            "lambda"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
        "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
        "lifecycleStage": "resolved",
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 0,
```

```
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "warning",
        "type": "priority"
    }]
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous ListOrganizationRecommendationsà la section Référence des AWS CLI commandes.

`list-recommendation-resources`

L'exemple de code suivant montre comment utiliser`list-recommendation-resources`.

AWS CLI

Pour répertorier les ressources de recommandation

L'`list-recommendation-resources`exemple suivant répertorie toutes les ressources associées à une recommandation en fonction de son identifiant.

```
aws trustedadvisor list-recommendation-resources \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578
```

Sortie :

```
{
    "recommendationResourceSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010
            "id":
 "18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
            "awsResourceId": "webcms-dev-01",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
```

```
                "0": "14",
                "1": "123.12000000000002",
                "2": "webcms-dev-01",
                "3": "db.m6i.large",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:webcms-dev-01",
                "7": "20"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-bbb7-491a-833b-5773e9589578/
e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
            "id":
 "e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
            "awsResourceId": "aws-dev-db-stack-instance-1",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "29.52",
                "2": "aws-dev-db-stack-instance-1",
                "3": "db.t2.small",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:aws-dev-db-stack-
instance-1",
                "7": "1"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459
```

```
            "id":
 "31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459",
            "awsResourceId": "aws-awesome-apps-stack-db",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "114.48000000000002",
                "2": "aws-awesome-apps-stack-db",
                "3": "db.m6g.large",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:aws-awesome-apps-stack-
db",
                "7": "100"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        }
    ],
    "nextToken": "REDACTED"
}
```


Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous ListRecommendationResourcesà la section Référence des AWS CLI commandes.

`list-recommendations`

L'exemple de code suivant montre comment utiliser`list-recommendations`.

AWS CLI

Exemple 1 : Pour répertorier les recommandations

L'`list-recommendations`exemple suivant répertorie toutes les recommandations et n'inclut aucun filtre.

```
aws trustedadvisor list-recommendations
```

Sortie :

```
{
    "recommendationSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "name": "MFA Recommendation",
            "awsServices": [
                "iam"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
            "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
            "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
            "pillarSpecificAggregates": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            },
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 1,
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "error",
            "type": "standard"
        },
        {
            "arn":
 "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
            "name": "RDS clusters quota warning",
            "awsServices": [
                "rds"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
```

```
            "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
            "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
            "pillarSpecificAggregates": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            },
            "pillars": [
                "service_limits"
            ],
            "resourcesAggregates": {
                "errorCount": 0,
                "okCount": 3,
                "warningCount": 6
            },
            "source": "ta_check",
            "status": "warning",
            "type": "standard"
        }
    ],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

Exemple 2 : pour répertorier les recommandations à l'aide d'un filtre

L'`list-recommendations`exemple suivant répertorie les recommandations et inclut un filtre.

```
aws trustedadvisor list-recommendations \
    --aws-service iam \
    --max-items 100
```

Sortie :

```
{
    "recommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
        "name": "MFA Recommendation",
        "awsServices": [
```

```
            "iam"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
        "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
        "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 1,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "error",
        "type": "standard"
    }],
    "nextToken": "REDACTED"
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

Exemple 3 : pour répertorier les recommandations à l'aide d'un jeton de pagination

L'`list-recommendations`exemple suivant utilise le « NextToken » renvoyé par une requête précédente pour récupérer la page suivante de recommandations filtrées.

```
aws trustedadvisor list-recommendations \
    --aws-service rds \
    --max-items 100 \
    --starting-token <next-token>
```

Sortie :

```
{
```

```
    "recommendationSummaries": [{
        "arn":
 "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
        "name": "RDS clusters quota warning",
        "awsServices": [
            "rds"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
        "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
        "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "pillars": [
            "service_limits"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 3,
            "warningCount": 6
        },
        "source": "ta_check",
        "status": "warning",
        "type": "standard"
    }]
}
```

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous ListRecommendationsà la section Référence des AWS CLI commandes.

`update-organization-recommendation-lifecycle`

L'exemple de code suivant montre comment utiliser`update-organization-recommendation-lifecycle`.

AWS CLI

Pour mettre à jour le cycle de vie des recommandations d'une organisation

L'`update-organization-recommendation-lifecycle`exemple suivant met à jour le cycle de vie d'une recommandation d'organisation en fonction de son identifiant.

```
aws trustedadvisor update-organization-recommendation-lifecycle \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/96b5e5ca-7930-444c-90c6-06d386128100 \
    --lifecycle-stage dismissed \
    --update-reason-code not_applicable
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous UpdateOrganizationRecommendationLifecycleà la section Référence des AWS CLI commandes.

`update-recommendation-lifecycle`

L'exemple de code suivant montre comment utiliser`update-recommendation-lifecycle`.

AWS CLI

Pour mettre à jour le cycle de vie d'une recommandation

L'`update-recommendation-lifecycle`exemple suivant met à jour le cycle de vie d'une recommandation en fonction de son identifiant.

```
aws trustedadvisor update-recommendation-lifecycle \
    --recommendation-identifier arn:aws:trustedadvisor::000000000000:recommendation/861c9c6e-f169-405a-8b59-537a8caccd7a \
    --lifecycle-stage resolved \
    --update-reason-code valid_business_case
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Commencer à utiliser l'API Trusted Advisor dans le guide de l'utilisateur de AWS Trusted Advisor.

- Pour plus de détails sur l'API, reportez-vous UpdateRecommendationLifecycleà la section Référence des AWS CLI commandes.

Exemples d'autorisations vérifiées utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide des autorisations vérifiées.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-identity-source`

L'exemple de code suivant montre comment utiliser`create-identity-source`.

AWS CLI

Pour créer une source d'identité

L'`create-identity-source`exemple suivant crée une source d'identité qui vous permet de référencer les identités stockées dans le groupe d'utilisateurs Amazon Cognito spécifié. Ces identités sont disponibles dans les autorisations vérifiées en tant qu'entités de type`User`.

```
aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu de `config.txt` :

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
    }
}
```

Sortie :

```
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Pour plus d'informations sur les sources d'identité, consultez la section Utilisation des autorisations vérifiées par Amazon auprès des fournisseurs d'identité dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous CreateIdentitySourceà la section Référence des AWS CLI commandes.

`create-policy-store`

L'exemple de code suivant montre comment utiliser`create-policy-store`.

AWS CLI

Pour créer un magasin de politiques

L'`create-policy-store`exemple suivant crée un magasin de politiques dans la AWS région actuelle.

```
aws verifiedpermissions create-policy-store \
    --validation-settings "mode=STRICT"
```

Sortie :

```
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Pour plus d'informations sur les boutiques politiques, consultez les boutiques Amazon Verified Permissions Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyStoreà la section Référence des AWS CLI commandes.

`create-policy-template`

L'exemple de code suivant montre comment utiliser`create-policy-template`.

AWS CLI

Pour créer un modèle de politique

L'`create-policy-template`exemple suivant crée un modèle de politique avec une instruction contenant un espace réservé pour le principal.

```
aws verifiedpermissions create-policy-template \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu de `template1.txt` :

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
```

```
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

Pour plus d'informations sur les modèles de politiques, consultez les modèles de politiques Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyTemplateà la section Référence des AWS CLI commandes.

`create-policy`

L'exemple de code suivant montre comment utiliser`create-policy`.

AWS CLI

Exemple 1 : pour créer une politique statique

L'`create-policy`exemple suivant crée une politique statique dont le champ d'application spécifie à la fois un principal et une ressource.

```
aws verifiedpermissions create-policy \
    --definition file://definition1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu du fichier `definition1.txt` :

```
{
    "static": {
        "description":  "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
    }
}
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
```

```
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

Exemple 2 : créer une politique statique qui accorde l'accès à une ressource à tout le monde

L'`create-policy`exemple suivant crée une politique statique dont la portée ne spécifie qu'une ressource.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu du fichier `definition2.txt` :

```
{
    "static": {
        "description":  "Grant everyone access to the publicFolder Album",
        "statement": "permit(principal, action, resource in Album::\"publicFolder
\");"
    }
}
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:39:44.975897+00:00",
    "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
    "policyId": "PbfR73F8oh5MMfr9uRtFDB",
    "policyStoreId": "PSEXAMPLEabcdefg222222",
    "policyType": "STATIC",
    "resource": {
```

```
        "entityId": "publicFolder",
        "entityType": "Album"
    }
}
```

Exemple 3 : pour créer une politique liée à un modèle associée au modèle spécifié

L'`create-policy`exemple suivant crée une politique liée à un modèle à l'aide du modèle de stratégie spécifié et associe le principal spécifié à utiliser à la nouvelle politique liée au modèle.

```
aws verifiedpermissions create-policy \
    --definition file://definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu de `definition.txt` :

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
            "entityType": "User",
            "entityId": "alice"
        }
    }
}
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:49:51.490211+00:00",
    "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATE_LINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
```

```
}
```

Pour plus d'informations sur les politiques, consultez les politiques d'Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous CreatePolicyà la section Référence des AWS CLI commandes.

`delete-identity-source`

L'exemple de code suivant montre comment utiliser`delete-identity-source`.

AWS CLI

Pour supprimer une source d'identité

L'`delete-identity-source`exemple suivant supprime la source d'identité qui possède l'identifiant spécifié.

```
aws verifiedpermissions delete-identity-source \
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur les sources d'identité, consultez la section Utilisation des autorisations vérifiées par Amazon auprès des fournisseurs d'identité dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous DeleteIdentitySourceà la section Référence des AWS CLI commandes.

`delete-policy-store`

L'exemple de code suivant montre comment utiliser`delete-policy-store`.

AWS CLI

Pour supprimer un magasin de politiques

L'`delete-policy-store`exemple suivant supprime le magasin de politiques qui possède l'ID spécifié.

```
aws verifiedpermissions delete-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur les boutiques politiques, consultez les boutiques Amazon Verified Permissions Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyStoreà la section Référence des AWS CLI commandes.

`delete-policy-template`

L'exemple de code suivant montre comment utiliser`delete-policy-template`.

AWS CLI

Pour supprimer un modèle de politique

L'`delete-policy-template`exemple suivant supprime le modèle de politique dont l'identifiant est spécifié.

```
aws verifiedpermissions delete-policy \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur les modèles de politiques, consultez les modèles de politiques Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyTemplateà la section Référence des AWS CLI commandes.

`delete-policy`

L'exemple de code suivant montre comment utiliser`delete-policy`.

AWS CLI

Pour supprimer une politique statique ou liée à un modèle

L'`delete-policy`exemple suivant supprime la politique dont l'identifiant est spécifié.

```
aws verifiedpermissions delete-policy \
    --policy-id SPEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Cette commande ne produit aucun résultat.

Pour plus d'informations sur les politiques, consultez les politiques d'Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`get-identity-source`

L'exemple de code suivant montre comment utiliser`get-identity-source`.

AWS CLI

Pour récupérer des informations sur une source d'identité

L'`get-identity-source`exemple suivant affiche les détails de la source d'identité avec l'ID spécifié.

```
aws verifiedpermissions get-identity-source \
    --identity-source  ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "createdDate": "2023-06-12T22:27:49.150035+00:00",
    "details": {
        "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
        "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_1a2b3c4d5",
        "openIdIssuer": "COGNITO",
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5"
    },
    "identitySourceId": "ISEXAMPLEabcdefg111111",
```

```
    "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
}
```

Pour plus d'informations sur les sources d'identité, consultez la section Utilisation des autorisations vérifiées par Amazon auprès des fournisseurs d'identité dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous GetIdentitySourceà la section Référence des AWS CLI commandes.

`get-policy-store`

L'exemple de code suivant montre comment utiliser`get-policy-store`.

AWS CLI

Pour récupérer les informations relatives à un magasin de politiques

L'`get-policy-store`exemple suivant affiche les détails du magasin de politiques avec l'ID spécifié.

```
aws verifiedpermissions get-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-06-05T20:16:46.225598+00:00",
    "lastUpdatedDate": "2023-06-08T20:40:23.173691+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "validationSettings": { "mode": "OFF" }
}
```

Pour plus d'informations sur les boutiques politiques, consultez les boutiques Amazon Verified Permissions Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous GetPolicyStoreà la section Référence des AWS CLI commandes.

`get-policy-template`

L'exemple de code suivant montre comment utiliser`get-policy-template`.

AWS CLI

Pour récupérer les informations relatives à un modèle de politique

L'`get-policy-template`exemple suivant affiche les détails du modèle de politique avec l'ID spécifié.

```
aws verifiedpermissions get-policy-template \
    --policy-template-id  PTEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111",
    "statement": "permit(\n    principal in ?principal,\n    action == Action::
\"view\",\n    resource == Photo::\"VacationPhoto94.jpg\"\n);"
}
```

Pour plus d'informations sur les modèles de politiques, consultez les modèles de politiques Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous GetPolicyTemplateà la section Référence des AWS CLI commandes.

`get-policy`

L'exemple de code suivant montre comment utiliser`get-policy`.

AWS CLI

Pour récupérer les informations relatives à une politique

L'`get-policy`exemple suivant affiche les détails de la politique avec l'ID spécifié.

```
aws verifiedpermissions get-policy \
```

```
    --policy-id PSEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "definition": {
        "static": {
            "description": "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
            "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
        }
    },
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

Pour plus d'informations sur les politiques, consultez les politiques d'Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous GetPolicyà la section Référence des AWS CLI commandes.

`get-schema`

L'exemple de code suivant montre comment utiliser`get-schema`.

AWS CLI

Pour récupérer le schéma dans un magasin de politiques

L'`get-schema`exemple suivant affiche les détails du schéma dans le magasin de politiques spécifié.

```
aws verifiedpermissions get-schema \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "schema": "{\"MySampleNamespace\":{\"entityTypes\":{\"Employee\":{\"shape
\":{\"attributes\":{\"jobLevel\":{\"type\":\"Long\"},\"name\":{\"type\":\"String
\"}},\"type\":\"Record\"}}},\"actions\":{\"remoteAccess\":{\"appliesTo\":
{\"principalTypes\":[\"Employee\"]}}}}",
    "createdDate": "2023-06-14T17:47:13.999885+00:00",
    "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

Pour plus d'informations sur le schéma, consultez le schéma du magasin Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous GetSchemaà la section Référence des AWS CLI commandes.

`is-authorized-with-token`

L'exemple de code suivant montre comment utiliser`is-authorized-with-token`.

AWS CLI

Exemple 1 : Pour demander une décision d'autorisation pour une demande d'utilisateur (autoriser)

L'`is-authorized-with-token`exemple suivant demande une décision d'autorisation pour un utilisateur authentifié par Amazon Cognito. La demande utilise le jeton d'identité fourni par Cognito plutôt que le jeton d'accès. Dans cet exemple, la banque d'informations spécifiée est configurée pour renvoyer les principaux sous forme d'entités de type`CognitoUser`.

```
aws verifiedpermissions is-authorized-with-token \
    --action actionId="View",actionType="Action" \
    --resource entityId="vacationPhoto94.jpg",entityType="Photo" \
    --policy-store-id PSEXAMPLEabcdefg111111 \
```

```
    --identity-token "AbCdE12345...long.string...54321EdCbA"
```

Le magasin de politiques contient une politique avec l'instruction suivante qui accepte les identités provenant du groupe d'utilisateurs et de l'identifiant d'application Cognito spécifiés.

```
permit(
    principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0kalbmc",
    action,
    resource == Photo::"VacationPhoto94.jpg"
);
```

Sortie :

```
{
    "decision":"Allow",
    "determiningPolicies":[
        {
        "determiningPolicyId":"SPEXAMPLEabcdefg111111"
        }
    ],
    "errors":[]
}
```

Pour plus d'informations sur l'utilisation des identités issues d'un groupe d'utilisateurs de Cognito, consultez la section Utilisation des autorisations vérifiées par Amazon avec les fournisseurs d'identité dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous IsAuthorizedWithTokenà la section Référence des AWS CLI commandes.

`is-authorized`

L'exemple de code suivant montre comment utiliser`is-authorized`.

AWS CLI

Exemple 1 : Pour demander une décision d'autorisation pour une demande d'utilisateur (autoriser)

L'`is-authorized`exemple suivant demande une décision d'autorisation pour un principal de type `User` named`Alice`, qui souhaite effectuer l'`updatePhoto`opération, sur une ressource de type `Photo` nommé`VacationPhoto94.jpg`.

La réponse indique que la demande est autorisée par une politique.

```
aws verifiedpermissions is-authorized \
    --principal entityType=User,entityId=alice \
    --action actionType=Action,actionId=view \
    --resource entityType=Photo,entityId=VactionPhoto94.jpg \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "decision": "ALLOW",
    "determiningPolicies": [
        {
            "policyId": "SPEXAMPLEabcdefg111111"
        }
    ],
    "errors": []
}
```

Exemple 2 : Pour demander une décision d'autorisation pour une demande utilisateur (refuser)

L'exemple suivant est identique à l'exemple précédent, sauf que le principal est`User::"Bob"`. Le magasin de politiques ne contient aucune politique autorisant cet utilisateur à accéder à`Album::"alice_folder"`.

La sortie indique que c'`Deny`était implicite car la liste des `DeterminingPolicies` est vide.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "decision": "DENY",
    "determiningPolicies": [],
    "errors": []
}
```

Pour plus d'informations, consultez le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous IsAuthorizedà la section Référence des AWS CLI commandes.

`list-identity-sources`

L'exemple de code suivant montre comment utiliser`list-identity-sources`.

AWS CLI

Pour répertorier les sources d'identité disponibles

L'`list-identity-sources`exemple suivant répertorie toutes les sources d'identité du magasin de politiques spécifié.

```
aws verifiedpermissions list-identity-sources \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "identitySources": [
        {
            "createdDate": "2023-06-12T22:27:49.150035+00:00",
            "details": {
                "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
                "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_1a2b3c4d5",
                "openIdIssuer": "COGNITO",
                "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5"
            },
            "identitySourceId": "ISEXAMPLEabcdefg111111",
            "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "principalEntityType": "User"
        }
    ]
}
```

Pour plus d'informations sur les sources d'identité, consultez la section Utilisation des autorisations vérifiées par Amazon auprès des fournisseurs d'identité dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous ListIdentitySourcesà la section Référence des AWS CLI commandes.

`list-policies`

L'exemple de code suivant montre comment utiliser`list-policies`.

AWS CLI

Pour répertorier les politiques disponibles

L'`list-policies`exemple suivant répertorie toutes les politiques du magasin de politiques spécifié.

```
aws verifiedpermissions list-policies \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "policies": [
        {
            "createdDate": "2023-06-12T20:33:37.382907+00:00",
            "definition": {
                "static": {
                    "description": "Grant everyone of janeFriends UserGroup access
 to the vacationFolder Album"
                }
            },
            "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
            "policyId": "SPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "principal": {
                "entityId": "janeFriends",
                "entityType": "UserGroup"
            },
            "resource": {
                "entityId": "vacationFolder",
                "entityType": "Album"
            }
        },
```

```
        {
            "createdDate": "2023-06-12T20:39:44.975897+00:00",
            "definition": {
                "static": {
                    "description": "Grant everyone access to the publicFolder Album"
                }
            },
            "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
            "policyId": "SPEXAMPLEabcdefg222222",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "resource": {
                "entityId": "publicFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-06-12T20:49:51.490211+00:00",
            "definition": {
                "templateLinked": {
                    "policyTemplateId": "PTEXAMPLEabcdefg111111"
                }
            },
            "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
            "policyId": "SPEXAMPLEabcdefg333333",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "TEMPLATE_LINKED",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "VacationPhoto94.jpg",
                "entityType": "Photo"
            }
        }
    ]
}
```


Pour plus d'informations sur les politiques, consultez les politiques d'Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous ListPoliciesà la section Référence des AWS CLI commandes.

`list-policy-stores`

L'exemple de code suivant montre comment utiliser`list-policy-stores`.

AWS CLI

Pour répertorier les magasins de politiques disponibles

L'`list-policy-stores`exemple suivant répertorie tous les magasins de politiques de la AWS région. Toutes les commandes relatives aux autorisations vérifiées, `list-policy-stores` sauf `create-policy-store` que vous devez spécifier l'ID du magasin de politiques avec lequel vous souhaitez travailler.

```
aws verifiedpermissions list-policy-stores
```

Sortie :

```
{
    "policyStores": [
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
            "createdDate": "2023-06-05T20:16:46.225598+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111"
        },
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg222222",
            "createdDate": "2023-06-08T18:09:37.364356+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg222222"
        },
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg333333",
            "createdDate": "2023-06-08T18:09:46.920600+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg333333"
        }
    ]
}
```

Pour plus d'informations sur les boutiques politiques, consultez les boutiques Amazon Verified Permissions Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous ListPolicyStoresà la section Référence des AWS CLI commandes.

`list-policy-templates`

L'exemple de code suivant montre comment utiliser`list-policy-templates`.

AWS CLI

Pour répertorier les modèles de politiques disponibles

L'`list-policy-templates`exemple suivant répertorie tous les modèles de politiques du magasin de politiques spécifié.

```
aws verifiedpermissions list-policy-templates \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "policyTemplates": [
        {
            "createdDate": "2023-06-12T20:47:42.804511+00:00",
            "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyTemplateId": "PTEXAMPLEabcdefg111111"
        }
    ]
}
```

Pour plus d'informations sur les modèles de politiques, consultez les modèles de politiques Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous ListPolicyTemplatesà la section Référence des AWS CLI commandes.

`put-schema`

L'exemple de code suivant montre comment utiliser`put-schema`.

AWS CLI

Pour enregistrer un schéma dans un magasin de politiques

L'`put-schema`exemple suivant crée ou remplace le schéma dans le magasin de politiques spécifié.

Le `cedarJson` paramètre du fichier d'entrée prend une représentation sous forme de chaîne d'un objet JSON. Il contient des guillemets intégrés («) dans la paire de guillemets la plus externe. Pour ce faire, vous devez convertir le JSON en chaîne en faisant précéder tous les guillemets incorporés d'une barre oblique inverse («) et en combinant toutes les lignes en une seule ligne de texte sans sauts de ligne.

Des exemples de chaînes peuvent être affichés sur plusieurs lignes ici pour des raisons de lisibilité, mais l'opération nécessite que les paramètres soient soumis sous forme de chaînes d'une seule ligne.

aws verifiedpermissions put-schema --fichier de définition : //schema.txt -- policy-store-id PSEXAMPLEabcdefg111111

Contenu de `schema.txt` :

```
{
    "cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
            \"appliesTo\": {\"principalTypes\": [\"Employee\"]}}},\"entityTypes\": {
            \"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\":
            \"Long\"},\"name\": {\"type\": \"String\"}},\"type\": \"Record\"}}}}}"
}
```

Sortie :

```
{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [
        "MySampleNamespace"
    ],
    "createdDate": "2023-06-14T17:47:13.999885+00:00",
    "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

Pour plus d'informations sur le schéma, consultez le schéma du magasin Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous PutSchemaà la section Référence des AWS CLI commandes.

`update-identity-source`

L'exemple de code suivant montre comment utiliser`update-identity-source`.

AWS CLI

Pour mettre à jour une source d'identité

L'`update-identity-source`exemple suivant modifie la source d'identité spécifiée en fournissant une nouvelle configuration du groupe d'utilisateurs Cognito et en modifiant le type d'entité renvoyé par la source d'identité.

```
aws verifiedpermissions update-identity-source
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --update-configuration file://config.txt \
    --principal-entity-type "Employee" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu de `config.txt` :

```
{
        "cognitoUserPoolConfiguration": {
                "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5",
                "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
        }
}
```

Sortie :

```
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Pour plus d'informations sur les sources d'identité, consultez la section Utilisation des autorisations vérifiées par Amazon auprès des fournisseurs d'identité dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous UpdateIdentitySourceà la section Référence des AWS CLI commandes.

`update-policy-store`

L'exemple de code suivant montre comment utiliser`update-policy-store`.

AWS CLI

Pour mettre à jour un magasin de politiques

L'`update-policy-store`exemple suivant modifie un magasin de politiques en modifiant son paramètre de validation.

```
aws verifiedpermissions update-policy-store \
    --validation-settings "mode=STRICT" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Sortie :

```
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Pour plus d'informations sur les boutiques politiques, consultez les boutiques Amazon Verified Permissions Policy dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous UpdatePolicyStoreà la section Référence des AWS CLI commandes.

`update-policy-template`

L'exemple de code suivant montre comment utiliser`update-policy-template`.

AWS CLI

Exemple 1 : pour mettre à jour un modèle de politique

L'`update-policy-template`exemple suivant modifie la politique liée au modèle spécifiée pour remplacer sa déclaration de politique.

```
aws verifiedpermissions update-policy-template \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Contenu du fichier `template1.txt` :

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

Pour plus d'informations sur les modèles de politiques, consultez les modèles de politiques Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous UpdatePolicyTemplateà la section Référence des AWS CLI commandes.

`update-policy`

L'exemple de code suivant montre comment utiliser`update-policy`.

AWS CLI

Pour mettre à jour une politique statique

L'`update-policy`exemple suivant modifie une politique statique existante en mettant à jour sa description et son instruction.

```
aws verifiedpermissions update-policy \
    --policy-id SPEXAMPLEabcdefg111111 \
    --definition file://updated-definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Le `statement` paramètre prend une représentation sous forme de chaîne d'un objet JSON. Il contient des guillemets intégrés («) dans la paire de guillemets la plus externe. Pour ce faire, vous devez convertir le JSON en chaîne en faisant précéder tous les guillemets incorporés d'une barre oblique inverse («) et en combinant toutes les lignes en une seule ligne de texte sans sauts de ligne.

Vous pouvez afficher des exemples de chaînes réparties sur plusieurs lignes pour plus de lisibilité, mais l'opération nécessite que les paramètres soient soumis sous forme de chaînes d'une seule ligne.

Contenu du fichier `updated-definition.txt` :

```
{
    "static": {
        "description": "Updated policy to grant janeFriends UserGroup access to the
 vacationFolder Album with view action only",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action ==
 Action::\"view\", resource in Album::\"vacationFolder\" );"
    }
}
```

Sortie :

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
```

```
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

Pour plus d'informations sur les politiques, consultez les politiques d'Amazon Verified Permissions dans le guide de l'utilisateur Amazon Verified Permissions.

- Pour plus de détails sur l'API, reportez-vous UpdatePolicyà la section Référence des AWS CLI commandes.

Exemples de treillis VPC utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface treillis avec VPC.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-listener`

L'exemple de code suivant montre comment utiliser`create-listener`.

AWS CLI

Pour créer un écouteur

L'`create-listener`exemple suivant crée un écouteur HTTPS avec une règle par défaut qui transmet le trafic au groupe cible VPC Lattice spécifié.

```
aws vpc-lattice create-listener \
    --name my-service-listener \
    --protocol HTTPS \
    --port 443 \
    --service-identifier svc-0285b53b2eEXAMPLE \
    --default-action file://listener-config.json
```

Contenu de `listener-config.json` :

```
{
    "forward": {
        "targetGroups": [
            {
                "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"
            }
        ]
    }
}
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-07cc7fb0abEXAMPLE",
    "defaultAction": {
        "forward": {
            "targetGroups": [
                {
                    "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE",
                    "weight": 100
                }
            ]
        }
    },
    "id": "listener-07cc7fb0abEXAMPLE",
    "name": "my-service-listener",
    "port": 443,
    "protocol": "HTTPS",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE"
}
```

Pour plus d'informations, consultez la section Listeners dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateListeneràla section Référence des AWS CLI commandes.

`create-resource-configuration`

L'exemple de code suivant montre comment utiliser`create-resource-configuration`.

AWS CLI

Pour créer une configuration de ressources

L'`create-resource-configuration`exemple suivant crée une configuration de ressource qui spécifie une IPv4 adresse unique.

```
aws vpc-lattice create-resource-configuration \
    --name my-resource-config \
    --type SINGLE \
    --resource-gateway-identifier rgw-0bba03f3d56060135 \
    --resource-configuration-definition 'ipResource={ipAddress=10.0.14.85}'
```

Sortie :

```
{
    "allowAssociationToShareableServiceNetwork": true,
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourceconfiguration/
rcfg-07129f3acded87625",
    "id": "rcfg-07129f3acded87625",
    "name": "my-resource-config",
    "portRanges": [
        "1-65535"
    ],
    "protocol": "TCP",
    "resourceConfigurationDefinition": {
        "ipResource": {
            "ipAddress": "10.0.14.85"
        }
    },
    "resourceGatewayId": "rgw-0bba03f3d56060135",
    "status": "ACTIVE",
```

```
    "type": "SINGLE"
}
```

Pour plus d'informations, consultez la section Configurations des ressources pour les ressources VPC dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateResourceConfigurationà la section Référence des AWS CLI commandes.

`create-resource-gateway`

L'exemple de code suivant montre comment utiliser`create-resource-gateway`.

AWS CLI

Pour créer une passerelle de ressources

L'`create-resource-gateway`exemple suivant crée une passerelle de ressources pour le sous-réseau spécifié.

```
aws vpc-lattice create-resource-gateway \
    --name my-resource-gateway \
    --vpc-identifier vpc-0bf4c2739bc05a69 \
    --subnet-ids subnet-08e8943905b63a683
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "id": "rgw-0bba03f3d56060135",
    "ipAddressType": "IPV4",
    "name": "my-resource-gateway",
    "securityGroupIds": [
        "sg-087ffd596c5fe962c"
    ],
    "status": "ACTIVE",
    "subnetIds": [
        "subnet-08e8943905b63a683"
    ],
    "vpcIdentifier": "vpc-0bf4c2739bc05a694"
}
```

Pour plus d'informations, consultez Resource gateway in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateResourceGatewayà la section Référence des AWS CLI commandes.

`create-service-network-service-association`

L'exemple de code suivant montre comment utiliser`create-service-network-service-association`.

AWS CLI

Pour créer une association de services

L'`create-service-network-service-association`exemple suivant associe le service spécifié au réseau de service spécifié.

```
aws vpc-lattice create-service-network-service-association \
    --service-identifier svc-0285b53b2eEXAMPLE \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-0e16955a8cEXAMPLE",
    "createdBy": "123456789012",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CEXAMPLE"
    },
    "id": "snsa-0e16955a8cEXAMPLE",
    "status": "CREATE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez Gérer les associations de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateServiceNetworkServiceAssociationà la section Référence des AWS CLI commandes.

`create-service-network-vpc-association`

L'exemple de code suivant montre comment utiliser`create-service-network-vpc-association`.

AWS CLI

Pour créer une association VPC

L'`create-service-network-vpc-association`exemple suivant associe le vpc spécifié au réseau de service spécifié. Le groupe de sécurité spécifié contrôle les ressources du VPC qui peuvent accéder au réseau de services et à ses services.

```
aws vpc-lattice create-service-network-vpc-association \
    --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --security-group-ids sg-0aee16bc6cEXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "createdBy": "123456789012",
    "id": "snva-0821fc8631EXAMPLE",
    "securityGroupIds": [
        "sg-0aee16bc6cEXAMPLE"
    ],
    "status": "CREATE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez Gérer les associations VPC dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateServiceNetworkVpcAssociationà la section Référence des AWS CLI commandes.

`create-service-network`

L'exemple de code suivant montre comment utiliser`create-service-network`.

AWS CLI

Pour créer un réseau de service

L'`create-service-network`exemple suivant crée un réseau de service portant le nom spécifié.

```
aws vpc-lattice create-service-network \
    --name my-service-network
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "authType": "NONE",
    "id": "sn-080ec7dc93EXAMPLE",
    "name": "my-service-network"
}
```

Pour plus d'informations, consultez la section Réseaux de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateServiceNetworkà la section Référence des AWS CLI commandes.

`create-service`

L'exemple de code suivant montre comment utiliser`create-service`.

AWS CLI

Pour créer un service

L'`create-service`exemple suivant crée un service portant le nom spécifié.

```
aws vpc-lattice create-service \
    --name my-lattice-service
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "authType": "NONE",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.1a2b3c4.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CEXAMPLE"
    },
    "id": "svc-0285b53b2eEXAMPLE",
    "name": "my-lattice-service",
    "status": "CREATE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez la section Services in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateServiceà la section Référence des AWS CLI commandes.

`create-target-group`

L'exemple de code suivant montre comment utiliser`create-target-group`.

AWS CLI

Exemple 1 : pour créer un groupe cible de type INSTANCE

L'`create-target-group`exemple suivant crée un groupe cible avec le nom, le type et la configuration spécifiés.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-instance \
    --type INSTANCE \
    --config file://tg-config.json
```

Contenu de `tg-config.json` :

```
{
    "port": 443,
    "protocol": "HTTPS",
```

```
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "healthCheck": {
            "enabled": true,
            "healthCheckIntervalSeconds": 30,
            "healthCheckTimeoutSeconds": 5,
            "healthyThresholdCount": 5,
            "matcher": {
                "httpCode": "200"
            },
            "path": "/",
            "protocol": "HTTPS",
            "protocolVersion": "HTTP1",
            "unhealthyThresholdCount": 2
        },
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-instance",
    "status": "CREATE_IN_PROGRESS",
    "type": "INSTANCE"
}
```

Exemple 2 : pour créer un groupe cible de type IP

L'`create-target-group`exemple suivant crée un groupe cible avec le nom, le type et la configuration spécifiés.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-ip \
    --type IP \
```

```
    --config file://tg-config.json
```

Contenu de `tg-config.json` :

```
{
    "ipAddressType": "IPV4",
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "healthCheck": {
            "enabled": true,
            "healthCheckIntervalSeconds": 30,
            "healthCheckTimeoutSeconds": 5,
            "healthyThresholdCount": 5,
            "matcher": {
                "httpCode": "200"
            },
            "path": "/",
            "protocol": "HTTPS",
            "protocolVersion": "HTTP1",
            "unhealthyThresholdCount": 2
        },
        "ipAddressType": "IPV4",
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-ip",
    "status": "CREATE_IN_PROGRESS",
    "type": "IP"
}
```

Exemple 3 : pour créer un groupe cible de type LAMBDA

L'`create-target-group`exemple suivant crée un groupe cible avec le nom, le type et la configuration spécifiés.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-lambda \
    --type LAMBDA
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-lambda",
    "status": "CREATE_IN_PROGRESS",
    "type": "LAMBDA"
}
```

Exemple 4 : Pour créer un groupe cible de type ALB

L'`create-target-group`exemple suivant crée un groupe cible avec le nom, le type et la configuration spécifiés.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-alb \
    --type ALB \
    --config file://tg-config.json
```

Contenu de `tg-config.json` :

```
{
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-alb",
    "status": "CREATE_IN_PROGRESS",
    "type": "ALB"
}
```

Pour plus d'informations, consultez la section Groupes cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous CreateTargetGroupà la section Référence des AWS CLI commandes.

`delete-auth-policy`

L'exemple de code suivant montre comment utiliser`delete-auth-policy`.

AWS CLI

Pour supprimer une politique d'authentification

L'`delete-auth-policy`exemple suivant supprime la politique d'authentification pour le service spécifié.

```
aws vpc-lattice delete-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les politiques d'authentification dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteAuthPolicyà la section Référence des AWS CLI commandes.

`delete-listener`

L'exemple de code suivant montre comment utiliser`delete-listener`.

AWS CLI

Pour supprimer un écouteur

L'`delete-listener`exemple suivant supprime l'écouteur spécifié.

```
aws vpc-lattice delete-listener \
    --listener-identifier listener-07cc7fb0abEXAMPLE \
    --service-identifier svc-0285b53b2eEXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Listeners dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteListenerà la section Référence des AWS CLI commandes.

`delete-resource-configuration`

L'exemple de code suivant montre comment utiliser`delete-resource-configuration`.

AWS CLI

Pour supprimer une configuration de ressource

L'`delete-resource-configuration`exemple suivant supprime la configuration de ressource spécifiée.

```
aws vpc-lattice delete-resource-configuration \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Resource gateway in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteResourceConfigurationà la section Référence des AWS CLI commandes.

`delete-resource-gateway`

L'exemple de code suivant montre comment utiliser`delete-resource-gateway`.

AWS CLI

Pour supprimer une passerelle de ressources

L'`delete-resource-gateway`exemple suivant supprime la passerelle de ressources spécifiée.

```
aws vpc-lattice delete-resource-gateway \
    --resource-gateway-identifier rgw-0bba03f3d56060135
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "id": "rgw-0bba03f3d56060135",
    "name": "my-resource-gateway",
    "status": "DELETE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez Resource gateway in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteResourceGatewayà la section Référence des AWS CLI commandes.

`delete-service-network-service-association`

L'exemple de code suivant montre comment utiliser`delete-service-network-service-association`.

AWS CLI

Pour supprimer une association de services

L'`delete-service-network-service-association`exemple suivant dissocie l'association de service spécifiée.

```
aws vpc-lattice delete-service-network-service-association \
```

```
    --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
    "id": "snsa-031fabb4d8EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez Gérer les associations de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceNetworkServiceAssociationà la section Référence des AWS CLI commandes.

`delete-service-network-vpc-association`

L'exemple de code suivant montre comment utiliser`delete-service-network-vpc-association`.

AWS CLI

Pour supprimer une association VPC

L'`delete-service-network-vpc-association`exemple suivant dissocie l'association VPC spécifiée.

```
aws vpc-lattice delete-service-network-vpc-association \
    --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "id": "snva-0821fc8631EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez Gérer les associations VPC dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceNetworkVpcAssociationà la section Référence des AWS CLI commandes.

`delete-service-network`

L'exemple de code suivant montre comment utiliser`delete-service-network`.

AWS CLI

Pour supprimer un réseau de service

L'`delete-service-network`exemple suivant supprime le réseau de service spécifié.

```
aws vpc-lattice delete-service-network \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Réseaux de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceNetworkà la section Référence des AWS CLI commandes.

`delete-service`

L'exemple de code suivant montre comment utiliser`delete-service`.

AWS CLI

Pour supprimer un service

L'`delete-service`exemple suivant supprime le service spécifié.

```
aws vpc-lattice delete-service \
    --service-identifier svc-0285b53b2eEXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "id": "svc-0285b53b2eEXAMPLE",
    "name": "my-lattice-service",
    "status": "DELETE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez la section Services in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteServiceà la section Référence des AWS CLI commandes.

`delete-target-group`

L'exemple de code suivant montre comment utiliser`delete-target-group`.

AWS CLI

Pour supprimer un groupe cible

L'`delete-target-group`exemple suivant supprime le groupe cible spécifié.

```
aws vpc-lattice delete-target-group \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

Pour plus d'informations, consultez la section Groupes cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeleteTargetGroupà la section Référence des AWS CLI commandes.

`deregister-targets`

L'exemple de code suivant montre comment utiliser`deregister-targets`.

AWS CLI

Pour désenregistrer une cible

L'`deregister-targets`exemple suivant désenregistre la cible spécifiée du groupe cible spécifié.

```
aws vpc-lattice deregister-targets \
    --targets i-07dd579bc5EXAMPLE \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Sortie :

```
{
    "successful": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443
        }
    ],
    "unsuccessful": []
}
```

Pour plus d'informations, consultez la section Enregistrer des cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous DeregisterTargetsà la section Référence des AWS CLI commandes.

`get-auth-policy`

L'exemple de code suivant montre comment utiliser`get-auth-policy`.

AWS CLI

Pour obtenir des informations sur une politique d'authentification

L'`get-auth-policy`exemple suivant permet d'obtenir des informations sur la politique d'authentification pour le service spécifié.

```
aws vpc-lattice get-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

Sortie :

```
{
    "createdAt": "2023-06-07T03:51:20.266Z",
    "lastUpdatedAt": "2023-06-07T04:39:27.082Z",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/my-clients\"},
\"Action\":\"vpc-lattice-svcs:Invoke\",\"Resource\":\"arn:aws:vpc-lattice:us-
east-2:123456789012:service/svc-0285b53b2eEXAMPLE\"}]}",
    "state": "Active"
}
```

Pour plus d'informations, consultez les politiques d'authentification dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetAuthPolicyà la section Référence des AWS CLI commandes.

`get-listener`

L'exemple de code suivant montre comment utiliser`get-listener`.

AWS CLI

Pour obtenir des informations sur un service listener

L'`get-listener`exemple suivant permet d'obtenir des informations sur l'écouteur spécifié pour le service spécifié.

```
aws vpc-lattice get-listener \
    --listener-identifier listener-0ccf55918cEXAMPLE \
    --service-identifier svc-0285b53b2eEXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
    "createdAt": "2023-05-07T05:08:45.192Z",
```

```
    "defaultAction": {
        "forward": {
            "targetGroups": [
                {
                    "targetGroupIdentifier": "tg-0ff213abb6EXAMPLE",
                    "weight": 1
                }
            ]
        }
    },
    "id": "listener-0ccf55918cEXAMPLE",
    "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
    "name": "http-80",
    "port": 80,
    "protocol": "HTTP",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE"
}
```

Pour plus d'informations, consultez la section Définir le routage dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetListenerà la section Référence des AWS CLI commandes.

`get-resource-configuration`

L'exemple de code suivant montre comment utiliser`get-resource-configuration`.

AWS CLI

Pour obtenir des informations sur la configuration d'une ressource

L'`get-resource-configuration`exemple suivant permet d'obtenir des informations sur la configuration de ressource spécifiée.

```
aws vpc-lattice get-resource-configuration \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

Sortie :

```
{
```

```
    "allowAssociationToShareableServiceNetwork": true,
    "amazonManaged": false,
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourceconfiguration/
rcfg-07129f3acded87625",
    "createdAt": "2025-02-01T00:57:35.871000+00:00",
    "id": "rcfg-07129f3acded87625",
    "lastUpdatedAt": "2025-02-01T00:57:46.874000+00:00",
    "name": "my-resource-config",
    "portRanges": [
        "1-65535"
    ],
    "protocol": "TCP",
    "resourceConfigurationDefinition": {
        "ipResource": {
            "ipAddress": "10.0.14.85"
        }
    },
    "resourceGatewayId": "rgw-0bba03f3d56060135",
    "status": "ACTIVE",
    "type": "SINGLE"
}
```

Pour plus d'informations, consultez Resource gateway in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetResourceConfigurationà la section Référence des AWS CLI commandes.

`get-resource-gateway`

L'exemple de code suivant montre comment utiliser`get-resource-gateway`.

AWS CLI

Pour obtenir des informations sur une passerelle de ressources

L'`get-resource-gateway`exemple suivant permet d'obtenir des informations sur la passerelle de ressources spécifiée.

```
aws vpc-lattice get-resource-gateway \
    --resource-gateway-identifier rgw-0bba03f3d56060135
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "createdAt": "2025-02-01T00:57:33.241000+00:00",
    "id": "rgw-0bba03f3d56060135",
    "ipAddressType": "IPV4",
    "lastUpdatedAt": "2025-02-01T00:57:44.351000+00:00",
    "name": "my-resource-gateway",
    "securityGroupIds": [
        "sg-087ffd596c5fe962c"
    ],
    "status": "ACTIVE",
    "subnetIds": [
        "subnet-08e8943905b63a683"
    ],
    "vpcId": "vpc-0bf4c2739bc05a694"
}
```

Pour plus d'informations, consultez Resource gateway in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetResourceGatewayà la section Référence des AWS CLI commandes.

`get-service-network-service-association`

L'exemple de code suivant montre comment utiliser`get-service-network-service-association`.

AWS CLI

Pour obtenir des informations sur une association de services

L'`get-service-network-service-association`exemple suivant permet d'obtenir des informations sur l'association de services spécifiée.

```
aws vpc-lattice get-service-network-service-association \
    --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

Sortie :

```
{
```

```
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
    "createdAt": "2023-05-05T21:48:16.076Z",
    "createdBy": "123456789012",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CEXAMPLE"
    },
    "id": "snsa-031fabb4d8EXAMPLE",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE",
    "serviceName": "my-lattice-service",
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "serviceNetworkId": "sn-080ec7dc93EXAMPLE",
    "serviceNetworkName": "my-service-network",
    "status": "ACTIVE"
}
```

Pour plus d'informations, consultez Gérer les associations de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetServiceNetworkServiceAssociationà la section Référence des AWS CLI commandes.

`get-service-network-vpc-association`

L'exemple de code suivant montre comment utiliser`get-service-network-vpc-association`.

AWS CLI

Pour obtenir des informations sur une association VPC

L'`get-service-network-vpc-association`exemple suivant obtient des informations sur l'association VPC spécifiée.

```
aws vpc-lattice get-service-network-vpc-association \
    --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "createdAt": "2023-06-06T23:41:08.421Z",
    "createdBy": "123456789012",
    "id": "snva-0c5dcb60d6EXAMPLE",
    "lastUpdatedAt": "2023-06-06T23:41:08.421Z",
    "securityGroupIds": [
        "sg-0aee16bc6cEXAMPLE"
    ],
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "serviceNetworkId": "sn-080ec7dc93EXAMPLE",
    "serviceNetworkName": "my-service-network",
    "status": "ACTIVE",
    "vpcId": "vpc-0a1b2c3d4eEXAMPLE"
}
```

Pour plus d'informations, consultez Gérer les associations VPC dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetServiceNetworkVpcAssociationà la section Référence des AWS CLI commandes.

`get-service-network`

L'exemple de code suivant montre comment utiliser`get-service-network`.

AWS CLI

Pour obtenir des informations sur un réseau de service

L'`get-service-network`exemple suivant permet d'obtenir des informations sur le réseau de service spécifié.

```
aws vpc-lattice get-service-network \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

Sortie :

```
{
```

```
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "authType": "AWS_IAM",
    "createdAt": "2023-05-05T15:26:08.417Z",
    "id": "sn-080ec7dc93EXAMPLE",
    "lastUpdatedAt": "2023-05-05T15:26:08.417Z",
    "name": "my-service-network",
    "numberOfAssociatedServices": 2,
    "numberOfAssociatedVPCs": 3
}
```

Pour plus d'informations, consultez la section Réseaux de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetServiceNetworkà la section Référence des AWS CLI commandes.

`get-service`

L'exemple de code suivant montre comment utiliser`get-service`.

AWS CLI

Pour obtenir des informations sur un service

L'`get-service`exemple suivant permet d'obtenir des informations sur le service spécifié.

```
aws vpc-lattice get-service \
    --service-identifier svc-0285b53b2eEXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "authType": "AWS_IAM",
    "createdAt": "2023-05-05T21:35:29.339Z",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CFUOHIZH"
    },
    "id": "svc-0285b53b2eEXAMPLE",
```

```
    "lastUpdatedAt": "2023-05-05T21:35:29.339Z",
    "name": "my-lattice-service",
    "status": "ACTIVE"
}
```

Pour plus d'informations, consultez la section Services du guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetServiceà la section Référence des AWS CLI commandes.

`get-target-group`

L'exemple de code suivant montre comment utiliser`get-target-group`.

AWS CLI

Pour obtenir des informations sur un groupe cible

L'`get-target-group`exemple suivant obtient des informations sur le groupe cible spécifié, dont le type de cible est`INSTANCE`.

```
aws vpc-lattice get-target-group \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Sortie :

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "healthCheck": {
            "enabled": true,
            "healthCheckIntervalSeconds": 30,
            "healthCheckTimeoutSeconds": 5,
            "healthyThresholdCount": 5,
            "matcher": {
                "httpCode": "200"
            },
            "path": "/",
            "protocol": "HTTPS",
            "protocolVersion": "HTTP1",
            "unhealthyThresholdCount": 2
```

```
        },
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "createdAt": "2023-05-06T04:41:04.122Z",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "lastUpdatedAt": "2023-05-06T04:41:04.122Z",
    "name": "my-target-group",
    "serviceArns": [
        "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE"
    ],
    "status": "ACTIVE",
    "type": "INSTANCE"
}
```

Pour plus d'informations, consultez la section Groupes cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous GetTargetGroupà la section Référence des AWS CLI commandes.

`list-listeners`

L'exemple de code suivant montre comment utiliser`list-listeners`.

AWS CLI

Pour répertorier les auditeurs du service

L'`list-listeners`exemple suivant répertorie les écouteurs pour le service spécifié.

```
aws vpc-lattice list-listeners \
    --service-identifier svc-0285b53b2eEXAMPLE
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
```

```
            "createdAt": "2023-05-07T05:08:45.192Z",
            "id": "listener-0ccf55918cEXAMPLE",
            "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
            "name": "http-80",
            "port": 80,
            "protocol": "HTTP"
        }
    ]
}
```

Pour plus d'informations, consultez la section Définir le routage dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListListenersà la section Référence des AWS CLI commandes.

`list-resource-configurations`

L'exemple de code suivant montre comment utiliser`list-resource-configurations`.

AWS CLI

Pour répertorier vos configurations de ressources

L'`list-resource-configurations`exemple suivant répertorie vos configurations de ressources.

```
aws vpc-lattice list-resource-configurations
```

Sortie :

```
{
    "items": [
        {
            "amazonManaged": false,
            "arn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625",
            "createdAt": "2025-02-01T00:57:35.871000+00:00",
            "id": "rcfg-07129f3acded87625",
            "lastUpdatedAt": "2025-02-01T00:57:46.874000+00:00",
            "name": "my-resource-config",
            "resourceGatewayId": "rgw-0bba03f3d56060135",
            "status": "ACTIVE",
```

```
            "type": "SINGLE"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configurations des ressources dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListResourceConfigurationsà la section Référence des AWS CLI commandes.

`list-resource-endpoint-associations`

L'exemple de code suivant montre comment utiliser`list-resource-endpoint-associations`.

AWS CLI

Pour répertorier les associations de points de terminaison VPC

L'`list-resource-endpoint-associations`exemple suivant répertorie les points de terminaison VPC associés à la configuration de ressources spécifiée.

```
aws vpc-lattice list-resource-endpoint-associations \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceendpointassociation/rea-0956a7435baf89326",
            "createdAt": "2025-02-01T00:57:38.998000+00:00",
            "id": "rea-0956a7435baf89326",
            "resourceConfigurationArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625",
            "resourceConfigurationId": "rcfg-07129f3acded87625",
            "vpcEndpointId": "vpce-019b90d6f16d4f958",
            "vpcEndpointOwner": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez Gérer les associations pour une configuration de ressources VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListResourceEndpointAssociationsà la section Référence des AWS CLI commandes.

`list-resource-gateways`

L'exemple de code suivant montre comment utiliser`list-resource-gateways`.

AWS CLI

Pour répertorier vos passerelles de ressources

L'`list-resource-gateways`exemple suivant répertorie vos passerelles de ressources.

```
aws vpc-lattice list-resource-gateways
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
            "createdAt": "2025-02-01T00:57:33.241000+00:00",
            "id": "rgw-0bba03f3d56060135",
            "ipAddressType": "IPV4",
            "lastUpdatedAt": "2025-02-01T00:57:44.351000+00:00",
            "name": "my-resource-gateway",
            "seurityGroupIds": [
                "sg-087ffd596c5fe962c"
            ],
            "status": "ACTIVE",
            "subnetIds": [
                "subnet-08e8943905b63a683"
            ],
            "vpcIdentifier": "vpc-0bf4c2739bc05a694"
        }
    ]
}
```

Pour plus d'informations, consultez Resource gateway in VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListResourceGatewaysà la section Référence des AWS CLI commandes.

`list-service-network-service-associations`

L'exemple de code suivant montre comment utiliser`list-service-network-service-associations`.

AWS CLI

Pour répertorier les associations de services

L'`list-service-network-service-associations`exemple suivant répertorie les associations de service pour le réseau de service spécifié. L'`--query`option étend la sortie à l'une IDs des associations de services.

```
aws vpc-lattice list-service-network-service-associations \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --query items[*].id
```

Sortie :

```
[
    "snsa-031fabb4d8EXAMPLE",
    "snsa-0e16955a8cEXAMPLE"
]
```

Pour plus d'informations, consultez Gérer les associations de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListServiceNetworkServiceAssociationsà la section Référence des AWS CLI commandes.

`list-service-network-vpc-associations`

L'exemple de code suivant montre comment utiliser`list-service-network-vpc-associations`.

AWS CLI

Pour répertorier les associations de VPC

L'`list-service-network-vpc-associations`exemple suivant répertorie les associations VPC pour le réseau de service spécifié. L'`--query`option étend la sortie à l'une IDs des associations VPC.

```
aws vpc-lattice list-service-network-vpc-associations \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --query items[*].id
```

Sortie :

```
[
    "snva-0821fc8631EXAMPLE",
    "snva-0c5dcb60d6EXAMPLE"
]
```

Pour plus d'informations, consultez Gérer les associations VPC dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListServiceNetworkVpcAssociationsà la section Référence des AWS CLI commandes.

`list-service-network-vpc-endpoint-associations`

L'exemple de code suivant montre comment utiliser`list-service-network-vpc-endpoint-associations`.

AWS CLI

Pour répertorier les associations de points de terminaison VPC

L'`list-service-network-vpc-endpoint-associations`exemple suivant répertorie les points de terminaison VPC associés au réseau de service spécifique.

```
aws vpc-lattice list-service-network-vpc-endpoint-associations \
    --service-network-identifier sn-0808d1748faee0c1e
```

Sortie :

```
{
    "items": [
        {
            "createdAt": "2025-02-01T01:21:36.667000+00:00",
            "serviceNetworkArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:servicenetwork/sn-0808d1748faee0c1e",
            "state": "ACTIVE",
            "vpcEndpointId": "vpce-0cc199f605eaeace7",
            "vpcEndpointOwnerId": "123456789012"
        }
    ]
}
```

Pour plus d'informations, consultez Gérer les associations pour un réseau de services VPC Lattice dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListServiceNetworkVpcEndpointAssociationsà la section Référence des AWS CLI commandes.

`list-service-networks`

L'exemple de code suivant montre comment utiliser`list-service-networks`.

AWS CLI

Pour répertorier vos réseaux de service

L'`list-service-networks`exemple suivant répertorie les réseaux de service détenus ou partagés avec le compte appelant. L'`--query`option étend les résultats aux Amazon Resource Names (ARN) des réseaux de services.

```
aws vpc-lattice list-service-networks \
    --query items[*].arn
```

Sortie :

```
[
    "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"
]
```

Pour plus d'informations, consultez la section Réseaux de services dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListServiceNetworksà la section Référence des AWS CLI commandes.

`list-services`

L'exemple de code suivant montre comment utiliser`list-services`.

AWS CLI

Pour répertorier vos services

L'`list-services`exemple suivant répertorie les services détenus ou partagés avec le compte d'appel. L'`--query`option étend les résultats aux Amazon Resource Names (ARN) des services.

```
aws vpc-lattice list-services \
    --query items[*].arn
```

Sortie :

```
[
    "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",
    "arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"
]
```

Pour plus d'informations, consultez la section Services du guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListServicesà la section Référence des AWS CLI commandes.

`list-target-groups`

L'exemple de code suivant montre comment utiliser`list-target-groups`.

AWS CLI

Pour répertorier vos groupes cibles

L'`list-target-groups`exemple suivant répertorie les groupes cibles dont le type de cible estLAMBDA.

```
aws vpc-lattice list-target-groups \
    --target-group-type LAMBDA
```

Sortie :

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-045c1b7d9dEXAMPLE",
            "createdAt": "2023-05-06T05:22:16.637Z",
            "id": "tg-045c1b7d9dEXAMPLE",
            "lastUpdatedAt": "2023-05-06T05:22:16.637Z",
            "name": "my-target-group-lam",
            "serviceArns": [
                "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE"
            ],
            "status": "ACTIVE",
            "type": "LAMBDA"
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListTargetGroupsà la section Référence des AWS CLI commandes.

`list-targets`

L'exemple de code suivant montre comment utiliser`list-targets`.

AWS CLI

Pour répertorier les cibles d'un groupe cible

L'`list-targets`exemple suivant répertorie les cibles pour le groupe cible spécifié.

```
aws vpc-lattice list-targets \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Sortie :

```
{
    "items": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443,
            "status": "HEALTHY"
        },
        {
            "id": "i-047b3c9078EXAMPLE",
            "port": 443,
            "reasonCode": "HealthCheckFailed",
            "status": "UNHEALTHY"
        }
    ]
}
```

Pour plus d'informations, consultez la section Groupes cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous ListTargetsà la section Référence des AWS CLI commandes.

`put-auth-policy`

L'exemple de code suivant montre comment utiliser`put-auth-policy`.

AWS CLI

Pour créer une politique d'authentification pour un service

L'`put-auth-policy`exemple suivant autorise l'accès aux demandes provenant de n'importe quel principal authentifié utilisant le rôle IAM spécifié. La ressource est l'ARN du service auquel la politique est attachée.

```
aws vpc-lattice put-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE \
```

```
    --policy file://auth-policy.json
```

Contenu de `auth-policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/my-clients"
            },
            "Action": "vpc-lattice-svcs:Invoke",
            "Resource": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE"
        }
    ]
}
```

Sortie :

```
{
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/my-clients\"},
\"Action\":\"vpc-lattice-svcs:Invoke\",\"Resource\":\"arn:aws:vpc-lattice:us-
east-2:123456789012:service/svc-0285b53b2eEXAMPLE\"}]}",
    "state": "Active"
}
```

Pour plus d'informations, consultez les politiques d'authentification dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous PutAuthPolicyà la section Référence des AWS CLI commandes.

`register-targets`

L'exemple de code suivant montre comment utiliser`register-targets`.

AWS CLI

Pour enregistrer une cible

L'`register-targets`exemple suivant enregistre les cibles spécifiées auprès du groupe cible spécifié.

```
aws vpc-lattice register-targets \
    --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Sortie :

```
{
    "successful": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443
        }
    ],
    "unsuccessful": [
        {
            "failureCode": "UnsupportedTarget",
            "failureMessage": "Instance targets must be in the same VPC as their
 target group",
            "id": "i-047b3c9078EXAMPLE",
            "port": 443
        }
    ]
}
```

Pour plus d'informations, consultez la section Enregistrer des cibles dans le guide de l'utilisateur Amazon VPC Lattice.

- Pour plus de détails sur l'API, reportez-vous RegisterTargetsà la section Référence des AWS CLI commandes.

AWS WAF Classic exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS WAF Classic.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`put-logging-configuration`

L'exemple de code suivant montre comment utiliser`put-logging-configuration`.

AWS CLI

Pour créer une configuration de journalisation pour l'ARN ACL Web avec l'ARN de flux Kinesis Firehose spécifié

L'`put-logging-configuration`exemple suivant montre la configuration de journalisation pour WAF avec CloudFront.

```
aws waf put-logging-configuration \
    --logging-configuration ResourceArn=arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[]
```

Sortie :

```
{
    "LoggingConfiguration": {
        "ResourceArn": "arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3",
        "LogDestinationConfigs": [
            "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream"
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutLoggingConfigurationà la section Référence des AWS CLI commandes.

update-byte-match-set

L'exemple de code suivant montre comment utiliserupdate-byte-match-set.

AWS CLI

Pour mettre à jour un jeu de correspondances d'octets

La update-byte-match-set commande suivante supprime un ByteMatchTuple objet (filtre) dans un ByteMatchSet :

```
aws waf update-byte-match-set --byte-match-set-id a123fae4-
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --
updates
 Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="b
```

Pour plus d'informations, consultez la section Travailler avec des conditions de correspondance de chaînes dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateByteMatchSetà la section Référence des AWS CLI commandes.

update-ip-set

L'exemple de code suivant montre comment utiliserupdate-ip-set.

AWS CLI

Pour mettre à jour un ensemble d'adresses IP

La update-ip-set commande suivante met à jour une IPSet IPv4 adresse et la supprime : IPv6

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="INSERT",IPSetDescriptor={Type="IPV4",Value="12.34.56.78/16"},Action="DELETE",IPSetD
```

Vous pouvez également utiliser un fichier JSON pour spécifier l'entrée. Par exemple :

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-
token 12cs345-67cd-890b-1cd2-c3a4567d89f1  --updates file://change.json
```

Où se trouve le contenu du fichier JSON :

```
[
{
"Action": "INSERT",
"IPSetDescriptor":
{
"Type": "IPV4",
"Value": "12.34.56.78/16"
}
},
{
"Action": "DELETE",
"IPSetDescriptor":
{
"Type": "IPV6",
"Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
}
}
]
```

Pour plus d'informations, consultez la section Travailler avec les conditions de correspondance IP dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateIpSetà la section Référence des AWS CLI commandes.

`update-rule`

L'exemple de code suivant montre comment utiliser`update-rule`.

AWS CLI

Pour mettre à jour une règle

La `update-rule` commande suivante supprime un objet Predicate dans une règle :

```
aws waf update-rule --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90
 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}
```

Pour plus d'informations, consultez la section Travailler avec des règles dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateRuleà la section Référence des AWS CLI commandes.

`update-size-constraint-set`

L'exemple de code suivant montre comment utiliser`update-size-constraint-set`.

AWS CLI

Pour mettre à jour un ensemble de contraintes de taille

La `update-size-constraint-set` commande suivante supprime un SizeConstraint objet (filtres) dans un ensemble de contraintes de taille :

```
aws waf update-size-constraint-set --size-constraint-set-id a123fae4-
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --
updates
 Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE
```

Pour plus d'informations, consultez la section Travailler avec des conditions de contrainte de taille dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateSizeConstraintSetà la section Référence des AWS CLI commandes.

`update-sql-injection-match-set`

L'exemple de code suivant montre comment utiliser`update-sql-injection-match-set`.

AWS CLI

Pour mettre à jour un ensemble de correspondances par injection SQL

La `update-sql-injection-match-set` commande suivante supprime un SqlInjectionMatchTuple objet (filtres) dans un jeu de correspondances par injection SQL :

```
aws waf update-sql-injection-match-set --sql-injection-
match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --
change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformati
```

Pour plus d'informations, consultez la section Travailler avec des conditions de correspondance par injection SQL dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateSqlInjectionMatchSetà la section Référence des AWS CLI commandes.

`update-web-acl`

L'exemple de code suivant montre comment utiliser`update-web-acl`.

AWS CLI

Pour mettre à jour une ACL Web

La `update-web-acl` commande suivante supprime un `ActivatedRule` objet dans un WebACL.

aws waf update-web-acl -- web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates Action="delete », =' {Priority=1, = » -1-Example », Action= {type="Autoriser «}, Type="Regular «} » ActivatedRule RuleId WAFRule

Sortie :

```
{
    "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
}
```

Pour plus d'informations, consultez Working with Web ACLs dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous UpdateWebAclà la section Référence des AWS CLI commandes.

`update-xss-match-set`

L'exemple de code suivant montre comment utiliser`update-xss-match-set`.

AWS CLI

Pour mettre à jour un XSSMatch ensemble

La `update-xss-match-set` commande suivante supprime un XssMatchTuple objet (filtres) dans un XssMatchSet :

```
aws waf update-xss-match-set --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_D
```

Pour plus d'informations, consultez la section Travailler avec les conditions de correspondance des scripts intersites dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateXssMatchSetà la section Référence des AWS CLI commandes.

AWS WAF Classic Regional exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS WAF Classic Regional.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-web-acl`

L'exemple de code suivant montre comment utiliser`associate-web-acl`.

AWS CLI

Pour associer une ACL Web à une ressource

La `associate-web-acl` commande suivante associe une ACL Web, spécifiée par le web-acl-id, à une ressource spécifiée par le resource-arn. L'ARN de la ressource peut faire référence à un équilibreur de charge d'application ou à une API Gateway :

```
aws waf-regional associate-web-acl \
    --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

Pour plus d'informations, consultez la section Travailler avec le Web ACLs dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous AssociateWebAclà la section Référence des AWS CLI commandes.

`put-logging-configuration`

L'exemple de code suivant montre comment utiliser`put-logging-configuration`.

AWS CLI

Pour créer une configuration de journalisation pour l'ARN ACL Web avec l'ARN de flux Kinesis Firehose spécifié

L'`put-logging-configuration`exemple suivant affiche la configuration de journalisation pour WAF ALB/APIGateway dans Region. `us-east-1`

```
aws waf-regional put-logging-configuration \
    --logging-configuration ResourceArn=arn:aws:waf-
regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-
a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-
east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[] \
    --region us-east-1
```

Sortie :

```
{
    "LoggingConfiguration": {
        "ResourceArn": "arn:aws:waf-regional:us-east-1:123456789012:webacl/3bffd3ed-
fa2e-445e-869f-a6a7cf153fd3",
        "LogDestinationConfigs": [
            "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-
firehose-stream"
```

```
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutLoggingConfigurationà la section Référence des AWS CLI commandes.

update-byte-match-set

L'exemple de code suivant montre comment utiliserupdate-byte-match-set.

AWS CLI

Pour mettre à jour un jeu de correspondances d'octets

La update-byte-match-set commande suivante supprime un ByteMatchTuple objet (filtre) dans unByteMatchSet. Comme la updates valeur comporte des guillemets doubles, vous devez l'entourer de guillemets simples.

```
aws waf-regional update-byte-match-set \
    --byte-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="
```

Pour plus d'informations, consultez la section Travailler avec des conditions de correspondance de chaînes dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateByteMatchSetà la section Référence des AWS CLI commandes.

update-ip-set

L'exemple de code suivant montre comment utiliserupdate-ip-set.

AWS CLI

Pour mettre à jour un ensemble d'adresses IP

La update-ip-set commande suivante met à jour une IPSet IPv4 adresse et la supprime. IPv6 Obtenez la valeur de change-token en exécutant la get-change-token commande.

Comme la valeur des mises à jour inclut des guillemets doubles intégrés, vous devez l'entourer de guillemets simples.

```
aws waf update-ip-set \
    --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="INSERT",IPSetDescriptor={Type="IPV4",Value="12.34.56.78/16"},Action="DELETE",IPSet
```

Vous pouvez également utiliser un fichier JSON pour spécifier l'entrée. Par exemple :

```
aws waf-regional update-ip-set \
    --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1  \
    --updates file://change.json
```

Contenu du `change.json`

```
[
    {
        "Action": "INSERT",
        "IPSetDescriptor":
        {
            "Type": "IPV4",
            "Value": "12.34.56.78/16"
        }
    },
    {
        "Action": "DELETE",
        "IPSetDescriptor":
        {
            "Type": "IPV6",
            "Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
        }
    }
]
```

Pour plus d'informations, consultez la section Travailler avec les conditions de correspondance IP dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateIpSetà la section Référence des AWS CLI commandes.

update-rule

L'exemple de code suivant montre comment utiliserupdate-rule.

AWS CLI

Pour mettre à jour une règle

La update-rule commande suivante supprime un Predicate objet dans une règle. Comme la updates valeur comporte des guillemets doubles, vous devez entourer la valeur entière de guillemets simples.

```
aws waf-regional update-rule \
    --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}'
```

Pour plus d'informations, consultez la section Travailler avec des règles dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateRuleà la section Référence des AWS CLI commandes.

update-size-constraint-set

L'exemple de code suivant montre comment utiliserupdate-size-constraint-set.

AWS CLI

Pour mettre à jour un ensemble de contraintes de taille

La update-size-constraint-set commande suivante supprime un objet SizeConstraint `(filtres) dans un ensemble de contraintes de taille. Comme la updates valeur contient des guillemets doubles incorporés, vous devez entourer la valeur entière de guillemets simples.

```
aws waf-regional update-size-constraint-set \
    --size-constraint-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NON
```

Pour plus d'informations, consultez la section Travailler avec des conditions de contrainte de taille dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateSizeConstraintSetà la section Référence des AWS CLI commandes.

`update-sql-injection-match-set`

L'exemple de code suivant montre comment utiliser`update-sql-injection-match-set`.

AWS CLI

Pour mettre à jour un ensemble de correspondances par injection SQL

La `update-sql-injection-match-set` commande suivante supprime un `SqlInjectionMatchTuple` objet (filtres) dans un jeu de correspondances par injection SQL. Comme la `updates` valeur contient des guillemets doubles intégrés, vous devez entourer la valeur entière de guillemets simples. :

aws waf-regional update-sql-injection-match -set -- sql-injection-match-set -id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates 'Action="delete », = {= {type="query_string "}, ="URL_DECODE"} 'SqlInjectionMatchTupleFieldToMatchTextTransformation

Pour plus d'informations, consultez la section Travailler avec des conditions de correspondance par injection SQL dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateSqlInjectionMatchSetà la section Référence des AWS CLI commandes.

`update-web-acl`

L'exemple de code suivant montre comment utiliser`update-web-acl`.

AWS CLI

Pour mettre à jour une ACL Web

La `update-web-acl` commande suivante supprime un `ActivatedRule` objet dans un WebACL. Comme la `updates` valeur contient des guillemets doubles intégrés, vous devez entourer la valeur entière de guillemets simples.

```
aws waf-regional update-web-acl \
    --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates Action="DELETE",ActivatedRule='{Priority=1,RuleId="WAFRule-1-
Example",Action={Type="ALLOW"},Type="ALLOW"}'
```

Pour plus d'informations, consultez la section Travailler avec le Web ACLs dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateWebAclà la section Référence des AWS CLI commandes.

`update-xss-match-set`

L'exemple de code suivant montre comment utiliser`update-xss-match-set`.

AWS CLI

Pour mettre à jour un XSSMatch ensemble

La `update-xss-match-set` commande suivante supprime un `XssMatchTuple` objet (filtres) dans un`XssMatchSet`. Comme la `updates` valeur contient des guillemets doubles incorporés, vous devez entourer la valeur entière de guillemets simples.

```
aws waf-regional update-xss-match-set \
    --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_
```

Pour plus d'informations, consultez la section Travailler avec les conditions de correspondance des scripts intersites dans le guide du développeur AWS WAF.

- Pour plus de détails sur l'API, reportez-vous UpdateXssMatchSetà la section Référence des AWS CLI commandes.

AWS WAFV2 exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with AWS WAFV2.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-web-acl`

L'exemple de code suivant montre comment utiliser`associate-web-acl`.

AWS CLI

Pour associer une ACL Web à une AWS ressource régionale

L'`associate-web-acl`exemple suivant associe l'ACL Web spécifiée à un Application Load Balancer.

```
aws wafv2 associate-web-acl \
    --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/waf-cli-alb/1ea17125f8b25a2a \
    --region us-west-2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Associer ou dissocier une ACL Web à une AWS ressource dans le AWS WAF, AWS Firewall Manager et AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous AssociateWebAclà la section Référence des AWS CLI commandes.

`check-capacity`

L'exemple de code suivant montre comment utiliser`check-capacity`.

AWS CLI

Pour obtenir la capacité utilisée par un ensemble de règles

Ce qui suit `check-capacity` permet de récupérer les exigences de capacité pour un ensemble de règles contenant une instruction de règle basée sur le taux et une instruction de règle AND contenant des règles imbriquées.

```
aws wafv2 check-capacity \
    --scope REGIONAL \
    --rules file://waf-rule-list.json \
    --region us-west-2
```

Contenu du fichier ://waf-rule-list.json :

```
[
    {
        "Name":"basic-rule",
        "Priority":0,
        "Statement":{
            "AndStatement":{
                "Statements":[
                    {
                        "ByteMatchStatement":{
                            "SearchString":"example.com",
                            "FieldToMatch":{
                                "SingleHeader":{
                                    "Name":"host"
                                }
                            },
                            "TextTransformations":[
                                {
                                    "Priority":0,
                                    "Type":"LOWERCASE"
                                }
                            ],
                            "PositionalConstraint":"EXACTLY"
                        }
                    },
                    {
                        "GeoMatchStatement":{
                            "CountryCodes":[
                                "US",
```

```
                                "IN"
                            ]
                        }
                    }
                ]
            }
        },
        "Action":{
            "Allow":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"basic-rule"
        }
    },
    {
        "Name":"rate-rule",
        "Priority":1,
        "Statement":{
            "RateBasedStatement":{
                "Limit":1000,
                "AggregateKeyType":"IP"
            }
        },
        "Action":{
            "Block":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"rate-rule"
        }
    }
]
```

Sortie :

```
{
```

```
    "Capacity":15
}
```

Pour plus d'informations, consultez la section AWS WAF Web ACL Capacity Units (WCU) dans le guide du AWS développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous CheckCapacityà la section Référence des AWS CLI commandes.

`create-ip-set`

L'exemple de code suivant montre comment utiliser`create-ip-set`.

AWS CLI

Pour créer un ensemble d'adresses IP à utiliser dans votre site Web ACLs et vos groupes de règles

La `create-ip-set` commande suivante crée un ensemble d'adresses IP avec une spécification de plage d'adresses unique.

```
aws wafv2 create-ip-set \
    --name testip \
    --scope REGIONAL \
    --ip-address-version IPV4 \
    --addresses 198.51.100.0/16
```

Sortie :

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"",
        "Name":"testip",
        "LockToken":"447e55ac-0000-0000-0000-86b67c17f8b5",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous CreateIpSetà la section Référence des AWS CLI commandes.

`create-regex-pattern-set`

L'exemple de code suivant montre comment utiliser`create-regex-pattern-set`.

AWS CLI

Pour créer un ensemble de modèles regex à utiliser dans votre site Web ACLs et vos groupes de règles

La `create-regex-pattern-set` commande suivante crée un ensemble de modèles regex avec deux modèles regex spécifiés.

```
aws wafv2 create-regex-pattern-set \
    --name regexPatterSet01 \
    --scope REGIONAL \
    --description 'Test web-acl' \
    --regular-expression-list '[{"RegexString": "/[0-9]*/"},{"RegexString": "/[a-
z]*/"}]'
```

Sortie :

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"Test web-acl",
        "Name":"regexPatterSet01",
        "LockToken":"0bc01e21-03c9-4b98-9433-6229cbf1ef1c",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous CreateRegexPatternSetà la section Référence des AWS CLI commandes.

`create-rule-group`

L'exemple de code suivant montre comment utiliser`create-rule-group`.

AWS CLI

Pour créer un groupe de règles personnalisé à utiliser sur votre site Web ACLs

La `create-rule-group` commande suivante crée un groupe de règles personnalisé pour une utilisation régionale. Les instructions de règle pour le groupe sont fournies dans un fichier au format JSON.

```
aws wafv2 create-rule-group \
    --name "TestRuleGroup" \
    --scope REGIONAL \
    --capacity 250 \
    --rules file://waf-rule.json \
    --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestRuleGroupMet
 \
    --region us-west-2
```

Contenu du fichier : //waf-rule.json :

```
[
    {
        "Name":"basic-rule",
        "Priority":0,
        "Statement":{
            "AndStatement":{
                "Statements":[
                    {
                        "ByteMatchStatement":{
                            "SearchString":"example.com",
                            "FieldToMatch":{
                                "SingleHeader":{
                                    "Name":"host"
                                }
                            },
                            "TextTransformations":[
                                {
                                    "Priority":0,
```

```
                                    "Type":"LOWERCASE"
                                }
                            ],
                            "PositionalConstraint":"EXACTLY"
                        }
                    },
                    {
                        "GeoMatchStatement":{
                            "CountryCodes":[
                                "US",
                                "IN"
                            ]
                        }
                    }
                ]
            }
        },
        "Action":{
            "Allow":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"basic-rule"
        }
    }
]
```

Sortie :

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"",
        "Name":"TestRuleGroup",
        "LockToken":"7b3bcec2-374e-4c5a-b2b9-563bf47249f0",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez Managing Your Own Rules Groups dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous CreateRuleGroupà la section Référence des AWS CLI commandes.

`create-web-acl`

L'exemple de code suivant montre comment utiliser`create-web-acl`.

AWS CLI

Pour créer une ACL Web

La `create-web-acl` commande suivante crée une ACL Web pour une utilisation régionale. Les instructions de règle pour l'ACL Web sont fournies dans un fichier au format JSON.

```
aws wafv2 create-web-acl \
    --name TestWebAcl \
    --scope REGIONAL \
    --default-action Allow={} \
    --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestWebAclMetric
 \
    --rules file://waf-rule.json \
    --region us-west-2
```

Contenu du fichier : //waf-rule.json :

```
[
    {
        "Name":"basic-rule",
        "Priority":0,
        "Statement":{
            "AndStatement":{
                "Statements":[
                    {
                        "ByteMatchStatement":{
                            "SearchString":"example.com",
                            "FieldToMatch":{
                                "SingleHeader":{
                                    "Name":"host"
```

```
                                }
                            },
                            "TextTransformations":[
                                {
                                    "Priority":0,
                                    "Type":"LOWERCASE"
                                }
                            ],
                            "PositionalConstraint":"EXACTLY"
                        }
                    },
                    {
                        "GeoMatchStatement":{
                            "CountryCodes":[
                                "US",
                                "IN"
                            ]
                        }
                    }
                ]
            }
        },
        "Action":{
            "Allow":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"basic-rule"
        }
    }
]
```

Sortie :

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"",
        "Name":"TestWebAcl",
```

```
        "LockToken":"2294b3a1-eb60-4aa0-a86f-a3ae04329de9",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

Pour plus d'informations, consultez Managing and Using a Web Access Control List (Web ACL) dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous CreateWebAclà la section Référence des AWS CLI commandes.

`delete-ip-set`

L'exemple de code suivant montre comment utiliser`delete-ip-set`.

AWS CLI

Pour supprimer un ensemble d'adresses IP

Ce qui suit `delete-ip-set` supprime l'ensemble d'adresses IP spécifié. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-ip-sets`, et un jeton de verrouillage, que vous pouvez obtenir à partir des appels, `list-ip-sets` et`get-ip-set`.

```
aws wafv2 delete-ip-set \
    --name test1 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 46851772-db6f-459d-9385-49428812e357
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteIpSetà la section Référence des AWS CLI commandes.

`delete-logging-configuration`

L'exemple de code suivant montre comment utiliser`delete-logging-configuration`.

AWS CLI

Pour désactiver la journalisation pour une ACL Web

Ce qui suit `delete-logging-configuration` supprime toute configuration de journalisation de l'ACL Web spécifiée.

```
aws wafv2 delete-logging-configuration \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Journalisation des informations relatives au trafic Web ACL dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteLoggingConfigurationà la section Référence des AWS CLI commandes.

`delete-regex-pattern-set`

L'exemple de code suivant montre comment utiliser`delete-regex-pattern-set`.

AWS CLI

Pour supprimer un ensemble de modèles regex

Ce qui suit `delete-regex-pattern-set` met à jour les paramètres du jeu de modèles regex spécifié. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-regex-pattern-sets`, et un jeton de verrouillage, que vous pouvez obtenir lors de l'appel `list-regex-pattern-sets` ou de l'appel`get-regex-pattern-set`.

```
aws wafv2 delete-regex-pattern-set \
    --name regexPatterSet01 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 0bc01e21-03c9-4b98-9433-6229cbf1ef1c
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteRegexPatternSetà la section Référence des AWS CLI commandes.

`delete-rule-group`

L'exemple de code suivant montre comment utiliser`delete-rule-group`.

AWS CLI

Pour supprimer un groupe de règles personnalisé

Ce qui suit `delete-rule-group` supprime le groupe de règles personnalisées spécifié. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-rule-groups`, et un jeton de verrouillage, que vous pouvez obtenir lors de l'appel `list-rule-groups` ou de l'appel`get-rule-group`.

```
aws wafv2 delete-rule-group \
    --name TestRuleGroup \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Managing Your Own Rules Groups dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteRuleGroupà la section Référence des AWS CLI commandes.

`delete-web-acl`

L'exemple de code suivant montre comment utiliser`delete-web-acl`.

AWS CLI

Pour supprimer une ACL Web

Ce qui suit `delete-web-acl` supprime l'ACL Web spécifié de votre compte. Une ACL Web ne peut être supprimée que lorsqu'elle n'est associée à aucune ressource. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-web-acls`, et un jeton de verrouillage, que vous pouvez obtenir lors de l'appel `list-web-acls` ou de l'appel`get-web-acl`.

```
aws wafv2 delete-web-acl \
    --name test \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token ebab4ed2-155e-4c9a-9efb-e4c45665b1f5
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Managing and Using a Web Access Control List (Web ACL) dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DeleteWebAclà la section Référence des AWS CLI commandes.

`describe-managed-rule-group`

L'exemple de code suivant montre comment utiliser`describe-managed-rule-group`.

AWS CLI

Pour récupérer la description d'un groupe de règles géré

La section suivante `describe-managed-rule-group` permet de récupérer la description d'un groupe de règles AWS géré.

```
aws wafv2 describe-managed-rule-group \
    --vendor-name AWS \
    --name AWSManagedRulesCommonRuleSet \
    --scope REGIONAL
```

Sortie :

```
{
    "Capacity": 700,
    "Rules": [
        {
            "Name": "NoUserAgent_HEADER",
```

```
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "UserAgent_BadBots_HEADER",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_QUERYSTRING",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_Cookie_HEADER",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_COOKIE",
            "Action": {
                "Block": {}
```

```
            }
        },
        {
            "Name": "EC2MetaDataSSRF_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericLFI_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            }
            "Name": "GenericLFI_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericLFI_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "RestrictedExtensions_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "RestrictedExtensions_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
```

```
        },
        {
            "Name": "GenericRFI_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_COOKIE",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_URIPATH",
            "Action": {
                "Block": {}
            }
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Groupes de règles gérés dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous DescribeManagedRuleGroupà la section Référence des AWS CLI commandes.

`disassociate-web-acl`

L'exemple de code suivant montre comment utiliser`disassociate-web-acl`.

AWS CLI

Pour dissocier une ACL Web d'une ressource régionale AWS

L'`disassociate-web-acl`exemple suivant supprime toute association ACL Web existante de l'Application Load Balancer spécifié.

```
aws wafv2 disassociate-web-acl \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a \
    --region us-west-2
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Associer ou dissocier une ACL Web à une AWS ressource dans le AWS WAF, AWS Firewall Manager et AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DisassociateWebAclà la section Référence des AWS CLI commandes.

`get-ip-set`

L'exemple de code suivant montre comment utiliser`get-ip-set`.

AWS CLI

Pour récupérer un ensemble d'adresses IP spécifique

Ce qui suit `get-ip-set` permet de récupérer l'ensemble d'adresses IP avec le nom, l'étendue et l'ID spécifiés. Vous pouvez obtenir l'ID d'un ensemble d'adresses IP à partir des commandes `create-ip-set` et`list-ip-sets`.

```
aws wafv2 get-ip-set \
    --name testip \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "IPSet":{
        "Description":"",
        "Name":"testip",
        "IPAddressVersion":"IPV4",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",
        "Addresses":[
            "192.0.2.0/16"
        ]
    },
    "LockToken":"447e55ac-2396-4c6d-b9f9-86b67c17f8b5"
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetIpSetà la section Référence des AWS CLI commandes.

`get-logging-configuration`

L'exemple de code suivant montre comment utiliser`get-logging-configuration`.

AWS CLI

Pour récupérer les configurations de journalisation pour une ACL Web

Ce qui suit `get-logging-configuration` récupère la configuration de journalisation pour l'ACL Web spécifiée.

```
aws wafv2 get-logging-configuration \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
    --region us-west-2
```

Sortie :

```
{
    "LoggingConfiguration":{
        "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "RedactedFields":[
            {
                "Method":{

                }
            }
        ],
        "LogDestinationConfigs":[
            "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-
custom-transformation"
        ]
    }
}
```

Pour plus d'informations, consultez la section Journalisation des informations relatives au trafic Web ACL dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetLoggingConfigurationà la section Référence des AWS CLI commandes.

`get-rate-based-statement-managed-keys`

L'exemple de code suivant montre comment utiliser`get-rate-based-statement-managed-keys`.

AWS CLI

Pour récupérer la liste des adresses IP bloquées par une règle basée sur le taux

Ce qui suit `get-rate-based-statement-managed-keys` permet de récupérer les adresses IP actuellement bloquées par une règle basée sur le taux utilisée pour une application régionale.

```
aws wafv2 get-rate-based-statement-managed-keys \
    --scope REGIONAL \
    --web-acl-name testwebacl2 \
    --web-acl-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --rule-name ratebasedtest
```

Sortie :

```
{
    "ManagedKeysIPV4":{
        "IPAddressVersion":"IPV4",
        "Addresses":[
            "198.51.100.0/32"
        ]
    },
    "ManagedKeysIPV6":{
        "IPAddressVersion":"IPV6",
        "Addresses":[

        ]
    }
}
```

Pour plus d'informations, consultez la section Déclaration de règle basée sur le taux dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetRateBasedStatementManagedKeysà la section Référence des AWS CLI commandes.

`get-regex-pattern-set`

L'exemple de code suivant montre comment utiliser`get-regex-pattern-set`.

AWS CLI

Pour récupérer un ensemble de modèles de regex spécifique

Ce qui suit `get-regex-pattern-set` récupère le modèle regex défini avec le nom, la portée, la région et l'ID spécifiés. Vous pouvez obtenir l'ID d'un ensemble de modèles regex à partir des commandes `create-regex-pattern-set` et`list-regex-pattern-sets`.

```
aws wafv2 get-regex-pattern-set \
    --name regexPatterSet01 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --region us-west-2
```

Sortie :

```
{
    "RegexPatternSet":{
        "Description":"Test web-acl",
        "RegularExpressionList":[
            {
                "RegexString":"/[0-9]*/"
            },
            {
                "RegexString":"/[a-z]*/"
            }
        ],
        "Name":"regexPatterSet01",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "LockToken":"c8abf33f-b6fc-46ae-846e-42f994d57b29"
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetRegexPatternSetà la section Référence des AWS CLI commandes.

`get-rule-group`

L'exemple de code suivant montre comment utiliser`get-rule-group`.

AWS CLI

Pour récupérer un groupe de règles personnalisé spécifique

Ce qui suit `get-rule-group` permet de récupérer le groupe de règles personnalisé avec le nom, la portée et l'ID spécifiés. Vous pouvez obtenir l'ID d'un groupe de règles à partir des commandes `create-rule-group` et`list-rule-groups`.

```
aws wafv2 get-rule-group \
    --name ff \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "RuleGroup":{
        "Capacity":1,
        "Description":"",
        "Rules":[
            {
                "Priority":0,
                "Action":{
                    "Block":{

                    }
                },
                "VisibilityConfig":{
                    "SampledRequestsEnabled":true,
                    "CloudWatchMetricsEnabled":true,
                    "MetricName":"jj"
                },
                "Name":"jj",
                "Statement":{
                    "SizeConstraintStatement":{
                        "ComparisonOperator":"LE",
                        "TextTransformations":[
                            {
                                "Priority":0,
                                "Type":"NONE"
                            }
                        ],
                        "FieldToMatch":{
```

```
                            "UriPath":{

                            }
                        },
                        "Size":7
                    }
                }
            }
        ],
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"ff"
        },
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/ff/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name":"ff"
    },
    "LockToken":"485458c9-1830-4234-af31-ec4d52ced1b3"
}
```

Pour plus d'informations, consultez Managing Your Own Rules Groups dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetRuleGroupà la section Référence des AWS CLI commandes.

`get-sampled-requests`

L'exemple de code suivant montre comment utiliser`get-sampled-requests`.

AWS CLI

Pour récupérer un échantillon de requêtes Web pour une ACL Web

Ce qui suit `get-sampled-requests` récupère les requêtes Web échantillonnées pour l'ACL Web, la métrique de règle et le délai spécifiés.

```
aws wafv2 get-sampled-requests \
    --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --rule-metric-name AWS-AWSManagedRulesSQLiRuleSet \
```

```
    --scope=REGIONAL \
    --time-window StartTime=2020-02-12T20:00Z,EndTime=2020-02-12T21:10Z \
    --max-items 100
```

Sortie :

```
{
    "TimeWindow": {
    "EndTime": 1581541800.0,
    "StartTime": 1581537600.0
    },
    "SampledRequests": [
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.564,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
```

```
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 1
        },
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.988,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 3
        },
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.846,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
```

```
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 1
        },
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.4,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
```

```
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 1
        }
    ],
    "PopulationSize": 4
}
```

Pour plus d'informations, consultez la section Visualisation d'un échantillon de requêtes Web dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetSampledRequestsà la section Référence des AWS CLI commandes.

`get-web-acl-for-resource`

L'exemple de code suivant montre comment utiliser`get-web-acl-for-resource`.

AWS CLI

Pour récupérer l'ACL Web associée à une AWS ressource

Ce qui suit `get-web-acl-for-resource` récupère le JSON pour l'ACL Web associée à la ressource spécifiée.

```
aws wafv2 get-web-acl-for-resource \
```

```
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/waf-cli-alb/1ea17125f8b25a2a
```

Sortie :

```
{
    "WebACL":{
        "Capacity":3,
        "Description":"",
        "Rules":[
            {
                "Priority":1,
                "Action":{
                    "Block":{

                    }
                },
                "VisibilityConfig":{
                    "SampledRequestsEnabled":true,
                    "CloudWatchMetricsEnabled":true,
                    "MetricName":"testrule01"
                },
                "Name":"testrule01",
                "Statement":{
                    "AndStatement":{
                        "Statements":[
                            {
                                "ByteMatchStatement":{
                                    "PositionalConstraint":"EXACTLY",
                                    "TextTransformations":[
                                        {
                                            "Priority":0,
                                            "Type":"NONE"
                                        }
                                    ],
                                    "SearchString":"dGVzdHN0cmluZw==",
                                    "FieldToMatch":{
                                        "UriPath":{

                                        }
                                    }
                                }
                            },
```

```
                            {
                                "SizeConstraintStatement":{
                                    "ComparisonOperator":"EQ",
                                    "TextTransformations":[
                                        {
                                            "Priority":0,
                                            "Type":"NONE"
                                        }
                                    ],
                                    "FieldToMatch":{
                                        "QueryString":{

                                       }
                                    },
                                    "Size":0
                                }
                            }
                        ]
                    }
                }
            }
        ],
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"test01"
        },
        "DefaultAction":{
            "Allow":{

            }
        },
        "Id":"9a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ",
        "Name":"test01"
    }
}
```

Pour plus d'informations, consultez la section Associer ou dissocier une ACL Web à une AWS ressource dans le AWS WAF, AWS Firewall Manager et AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetWebAclForResourceà la section Référence des AWS CLI commandes.

`get-web-acl`

L'exemple de code suivant montre comment utiliser`get-web-acl`.

AWS CLI

Pour récupérer une ACL Web

Ce qui suit `get-web-acl` récupère l'ACL Web avec le nom, la portée et l'ID spécifiés. Vous pouvez obtenir l'ID d'une ACL Web à partir des commandes `create-web-acl` et`list-web-acls`.

```
aws wafv2 get-web-acl \
    --name test01 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "WebACL":{
        "Capacity":3,
        "Description":"",
        "Rules":[
            {
                "Priority":1,
                "Action":{
                    "Block":{

                    }
                },
                "VisibilityConfig":{
                   "SampledRequestsEnabled":true,
                    "CloudWatchMetricsEnabled":true,
                    "MetricName":"testrule01"
                },
                "Name":"testrule01",
                "Statement":{
                    "AndStatement":{
                        "Statements":[
                            {
                                "ByteMatchStatement":{
                                    "PositionalConstraint":"EXACTLY",
```

```
                                "TextTransformations":[
                                    {
                                        "Priority":0,
                                        "Type":"NONE"
                                    }
                                ],
                                "SearchString":"dGVzdHN0cmluZw==",
                                "FieldToMatch":{
                                    "UriPath":{

                                    }
                                }
                            }
                        },
                        {
                            "SizeConstraintStatement":{
                                "ComparisonOperator":"EQ",
                                "TextTransformations":[
                                    {
                                        "Priority":0,
                                        "Type":"NONE"
                                    }
                                ],
                                "FieldToMatch":{
                                    "QueryString":{

                                    }
                                },
                                "Size":0
                            }
                        }
                    ]
                }
            }
        }
    ],
    "VisibilityConfig":{
        "SampledRequestsEnabled":true,
        "CloudWatchMetricsEnabled":true,
        "MetricName":"test01"
    },
    "DefaultAction":{
        "Allow":{
```

```
            }
        },
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name":"test01"
    },
    "LockToken":"e3db7e2c-d58b-4ee6-8346-6aec5511c6fb"
}
```

Pour plus d'informations, consultez Managing and Using a Web Access Control List (Web ACL) dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous GetWebAclà la section Référence des AWS CLI commandes.

`list-available-managed-rule-groups`

L'exemple de code suivant montre comment utiliser`list-available-managed-rule-groups`.

AWS CLI

Pour récupérer les groupes de règles gérés

Ce qui suit `list-available-managed-rule-groups` renvoie la liste de tous les groupes de règles gérés actuellement disponibles pour une utilisation sur votre site Web ACLs.

```
aws wafv2 list-available-managed-rule-groups \
    --scope REGIONAL
```

Sortie :

```
{
   "ManagedRuleGroups": [
       {
           "VendorName": "AWS",
           "Name": "AWSManagedRulesCommonRuleSet",
           "Description": "Contains rules that are generally applicable to web
applications. This provides protection against exploitation of a wide range of
vulnerabilities, including those described in OWASP publications and common Common
Vulnerabilities and Exposures (CVE)."
       },
       {
```

```
            "VendorName": "AWS",
            "Name": "AWSManagedRulesAdminProtectionRuleSet",
            "Description": "Contains rules that allow you to block external access
to exposed admin pages. This may be useful if you are running third-party software
or would like to reduce the risk of a malicious actor gaining administrative access
to your application."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesKnownBadInputsRuleSet",
            "Description": "Contains rules that allow you to block request patterns
that are known to be invalid and are associated with exploitation or discovery of
vulnerabilities. This can help reduce the risk of a malicious actor discovering a
vulnerable application."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesSQLiRuleSet",
            "Description": "Contains rules that allow you to block request patterns
associated with exploitation of SQL databases, like SQL injection attacks. This can
help prevent remote injection of unauthorized queries."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesLinuxRuleSet",
            "Description": "Contains rules that block request patterns associated
with exploitation of vulnerabilities specific to Linux, including LFI attacks. This
can help prevent attacks that expose file contents or execute code for which the
attacker should not have had access."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesUnixRuleSet",
            "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to POSIX/POSIX-like OS, including LFI
attacks. This can help prevent attacks that expose file contents or execute code
for which access should not been allowed."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesWindowsRuleSet",
            "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to Windows, (e.g., PowerShell commands).
```

```
This can help prevent exploits that allow attacker to run unauthorized commands or
execute malicious code."
       },
       {
           "VendorName": "AWS",
           "Name": "AWSManagedRulesPHPRuleSet",
           "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to the use of the PHP, including injection
of unsafe PHP functions. This can help prevent exploits that allow an attacker to
remotely execute code or commands."
       },
       {
           "VendorName": "AWS",
           "Name": "AWSManagedRulesWordPressRuleSet",
           "Description": "The WordPress Applications group contains rules that
block request patterns associated with the exploitation of vulnerabilities specific
to WordPress sites."
       },
       {
           "VendorName": "AWS",
           "Name": "AWSManagedRulesAmazonIpReputationList",
           "Description": "This group contains rules that are based on Amazon
threat intelligence. This is useful if you would like to block sources associated
with bots or other threats."
       }
   ]
}
```

Pour plus d'informations, consultez la section Groupes de règles gérés dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListAvailableManagedRuleGroupsà la section Référence des AWS CLI commandes.

`list-ip-sets`

L'exemple de code suivant montre comment utiliser`list-ip-sets`.

AWS CLI

Pour récupérer une liste d'ensembles d'adresses IP

Ce qui suit `list-ip-sets` permet de récupérer tous les ensembles d'adresses IP du compte qui ont une portée régionale.

```
aws wafv2 list-ip-sets \
    --scope REGIONAL
```

Sortie :

```
{
    "IPSets":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"",
            "Name":"testip",
            "LockToken":"0674c84b-0304-47fe-8728-c6bff46af8fc",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  "
        }
    ],
    "NextMarker":"testip"
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListIpSetsà la section Référence des AWS CLI commandes.

`list-logging-configurations`

L'exemple de code suivant montre comment utiliser`list-logging-configurations`.

AWS CLI

Pour récupérer la liste de toutes les configurations de journalisation pour une région

Ce qui suit `list-logging-configurations` permet de récupérer toutes les configurations de journalisation pour le Web ACLs qui sont adaptées à une utilisation régionale dans la `us-west-2` région.

```
aws wafv2 list-logging-configurations \
    --scope REGIONAL \
```

```
    --region us-west-2
```

Sortie :

```
{
    "LoggingConfigurations":[
        {
            "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
test-2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RedactedFields":[
                {
                    "QueryString":{

                    }
                }
            ],
            "LogDestinationConfigs":[
                "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-
logs-test"
            ]
        },
        {
            "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
test/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "RedactedFields":[
                {
                    "Method":{

                    }
                }
            ],
            "LogDestinationConfigs":[
                "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-
logs-custom-transformation"
            ]
        }
    ]
}
```

Pour plus d'informations, consultez la section Journalisation des informations relatives au trafic Web ACL dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListLoggingConfigurationsà la section Référence des AWS CLI commandes.

`list-regex-pattern-sets`

L'exemple de code suivant montre comment utiliser`list-regex-pattern-sets`.

AWS CLI

Pour récupérer une liste d'ensembles de modèles regex

Ce qui suit `list-regex-pattern-sets` permet de récupérer tous les ensembles de modèles regex pour le compte définis dans la région. `us-west-2`

```
aws wafv2 list-regex-pattern-sets \
--scope REGIONAL \
--region us-west-2
```

Sortie :

```
{
    "NextMarker":"regexPatterSet01",
    "RegexPatternSets":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"Test web-acl",
            "Name":"regexPatterSet01",
            "LockToken":"f17743f7-0000-0000-0000-19a8b93bfb01",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ]
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListRegexPatternSetsà la section Référence des AWS CLI commandes.

`list-resources-for-web-acl`

L'exemple de code suivant montre comment utiliser`list-resources-for-web-acl`.

AWS CLI

Pour récupérer les ressources associées à une ACL Web

Ce qui suit `list-resources-for-web-acl` permet de récupérer les ressources de l'API REST API Gateway actuellement associées à l'ACL Web spécifiée dans la région. `us-west-2`

```
aws wafv2 list-resources-for-web-acl \
    --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --resource-type API_GATEWAY \
    --region us-west-2
```

Sortie :

```
{
    "ResourceArns": [
        "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
    ]
}
```

Pour plus d'informations, consultez la section Associer ou dissocier une ACL Web à une AWS ressource dans le AWS WAF, AWS Firewall Manager et AWS Shield Advanced Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListResourcesForWebAclà la section Référence des AWS CLI commandes.

`list-rule-groups`

L'exemple de code suivant montre comment utiliser`list-rule-groups`.

AWS CLI

Pour récupérer une liste de groupes de règles personnalisés

Ce qui suit `list-rule-groups` permet de récupérer tous les groupes de règles personnalisés définis pour le compte pour l'étendue et l'emplacement de la région spécifiés.

```
aws wafv2 list-rule-groups \
    --scope REGIONAL \
    --region us-west-2
```

Sortie :

```
{
    "RuleGroups":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"",
            "Name":"TestRuleGroup",
            "LockToken":"1eb5ec48-0000-0000-0000-ee9b906c541e",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Description":"",
            "Name":"test",
            "LockToken":"b0f4583e-998b-4880-9069-3fbe45738b43",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
        }
    ],
    "NextMarker":"test"
}
```

Pour plus d'informations, consultez Managing Your Own Rules Groups dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListRuleGroupsà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour récupérer toutes les balises d'une ressource AWS WAF

Ce qui suit `list-tags-for-resource` permet de récupérer la liste de toutes les paires de balises, clés et valeurs pour l'ACL Web spécifiée.

```
aws wafv2 list-tags-for-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Sortie :

```
{
    "NextMarker":"",
    "TagInfoForResource":{
        "ResourceARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "TagList":[

        ]
    }
}
```

Pour plus d'informations, consultez Getting Started with AWS WAF dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-web-acls`

L'exemple de code suivant montre comment utiliser`list-web-acls`.

AWS CLI

Pour récupérer le Web ACLs pour une portée

Ce qui suit `list-web-acls` récupère tous ACLs les sites Web définis pour le compte pour la portée spécifiée.

```
aws wafv2 list-web-acls \
    --scope REGIONAL
```

Sortie :

```
{
    "NextMarker":"Testt",
    "WebACLs":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/Testt/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"sssss",
            "Name":"Testt",
            "LockToken":"7f36cb30-74ef-4cff-8cd4-a77e1aba1746",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ]
}
```

Pour plus d'informations, consultez Managing and Using a Web Access Control List (Web ACL) dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous ListWebAclsà la section Référence des AWS CLI commandes.

`put-logging-configuration`

L'exemple de code suivant montre comment utiliser`put-logging-configuration`.

AWS CLI

Pour ajouter une configuration de journalisation à une ACL Web

Ce qui suit `put-logging-configuration` ajoute la configuration de `aws-waf-logs-custom-transformation` journalisation d'Amazon Kinesis Data Firehose à l'ACL Web spécifiée, sans qu'aucun champ ne soit supprimé des journaux.

```
aws wafv2 put-logging-configuration \
    --logging-configuration ResourceArn=arn:aws:wafv2:us-
west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111,LogDestinationConfigs=arn:aws:firehose:us-
west-2:123456789012:deliverystream/aws-waf-logs-custom-transformation \
            --region us-west-2
```

Sortie :

```
{
```

```
    "LoggingConfiguration":{
        "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-
cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "LogDestinationConfigs":[
            "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-
custom-transformation"
        ]
    }
}
```

Pour plus d'informations, consultez la section Journalisation des informations relatives au trafic Web ACL dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous PutLoggingConfigurationà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour ajouter des balises à une ressource AWS WAF

L'`tag-resource`exemple suivant ajoute une balise dont la clé `Name` et la valeur sont définies sur `AWSWAF` l'ACL Web spécifiée.

```
aws wafv2 tag-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tags Key=Name,Value=AWSWAF
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting Started with AWS WAF dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour supprimer des balises d'une ressource AWS WAF

L'`untag-resource`exemple suivant supprime la balise contenant la clé `KeyName` de l'ACL Web spécifiée.

```
aws wafv2 untag-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tag-keys "KeyName"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Getting Started with AWS WAF dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-ip-set`

L'exemple de code suivant montre comment utiliser`update-ip-set`.

AWS CLI

Pour modifier les paramètres d'un ensemble d'adresses IP existant

Ce qui suit `update-ip-set` met à jour les paramètres de l'ensemble d'adresses IP spécifié. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-ip-sets`, et un jeton de verrouillage que vous pouvez obtenir lors des appels, `list-ip-sets` et`get-ip-set`. Cet appel renvoie également un jeton de verrouillage que vous pouvez utiliser pour une mise à jour ultérieure.

```
aws wafv2 update-ip-set \
    --name testip \
    --scope REGIONAL \
```

```
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --addresses 198.51.100.0/16 \
    --lock-token 447e55ac-2396-4c6d-b9f9-86b67c17f8b5
```

Sortie :

```
{
    "NextLockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc"
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous UpdateIpSetà la section Référence des AWS CLI commandes.

update-regex-pattern-set

L'exemple de code suivant montre comment utiliserupdate-regex-pattern-set.

AWS CLI

Pour modifier les paramètres d'un ensemble de modèles regex existant

Ce qui suit update-regex-pattern-set met à jour les paramètres du jeu de modèles regex spécifié. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appellist-regex-pattern-sets, et un jeton de verrouillage que vous pouvez obtenir lors des appels, list-regex-pattern-sets etget-regex-pattern-set. Cet appel renvoie également un jeton de verrouillage que vous pouvez utiliser pour une mise à jour ultérieure.

```
aws wafv2 update-regex-pattern-set \
    --name ExampleRegex \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --regular-expression-list RegexString="^.+$"  \
    --lock-token ed207e9c-82e9-4a77-aadd-81e6173ab7eb
```

Sortie :

```
{
```

```
    "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"
}
```

Pour plus d'informations, consultez les sections IP Sets et Regex Pattern Sets dans le AWS guide du développeur WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous UpdateRegexPatternSetà la section Référence des AWS CLI commandes.

`update-rule-group`

L'exemple de code suivant montre comment utiliser`update-rule-group`.

AWS CLI

Pour mettre à jour un groupe de règles personnalisé

Ce qui suit `update-rule-group` modifie la configuration de visibilité d'un groupe de règles personnalisé existant. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-rule-groups`, et un jeton de verrouillage que vous pouvez obtenir lors des appels, `list-rule-groups` et`get-rule-group`. Cet appel renvoie également un jeton de verrouillage que vous pouvez utiliser pour une mise à jour ultérieure.

```
aws wafv2 update-rule-group \
    --name TestRuleGroup \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0 \
    --visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=TestMetricsFor
 \
    --region us-west-2
```

Sortie :

```
{
    "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"
}
```

Pour plus d'informations, consultez Managing Your Own Rules Groups dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous UpdateRuleGroupà la section Référence des AWS CLI commandes.

`update-web-acl`

L'exemple de code suivant montre comment utiliser`update-web-acl`.

AWS CLI

Pour mettre à jour une ACL Web

Ce qui suit `update-web-acl` modifie les paramètres d'une ACL Web existante. Cet appel nécessite un identifiant, que vous pouvez obtenir lors de l'appel`list-web-acls`, un jeton de verrouillage et d'autres paramètres, que vous pouvez obtenir lors de l'appel`get-web-acl`. Cet appel renvoie également un jeton de verrouillage que vous pouvez utiliser pour une mise à jour ultérieure.

```
aws wafv2 update-web-acl \
    --name TestWebAcl \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 2294b3a1-0000-0000-0000-a3ae04329de9 \
    --default-action Block={} \
    --visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=NewMetricTestW
 \
    --rules file://waf-rule.json \
    --region us-west-2
```

Sortie :

```
{
    "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"
}
```

Pour plus d'informations, consultez Managing and Using a Web Access Control List (Web ACL) dans le guide du développeur AWS WAF, AWS Firewall Manager et AWS Shield Advanced.

- Pour plus de détails sur l'API, reportez-vous UpdateWebAclà la section Référence des AWS CLI commandes.

WorkDocs exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with WorkDocs.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`abort-document-version-upload`

L'exemple de code suivant montre comment utiliser`abort-document-version-upload`.

AWS CLI

Pour arrêter le téléchargement de la version d'un document

Cet exemple arrête le téléchargement d'une version de document déjà lancé.

Commande :

```
aws workdocs abort-document-version-upload --document-
id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f012673132098 35f2 --version-
id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous AbortDocumentVersionUploadà la section Référence des AWS CLI commandes.

`activate-user`

L'exemple de code suivant montre comment utiliser`activate-user`.

AWS CLI

Pour activer un utilisateur

Cet exemple active un utilisateur inactif.

Commande :

```
aws workdocs activate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Sortie :

```
{
  "User": {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "exampleUser",
      "EmailAddress": "exampleUser@site.awsapps.com",
      "GivenName": "Example",
      "Surname": "User",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
 "75f67c183aa1217409ac87576a45c03a5df5e6d8c51c35c01669970538e86cd0",
      "RecycleBinFolderId":
 "642b7dd3e60b14204534f3df7b1959e01b5d170f8c2707f410e40a8149120a57",
      "Status": "ACTIVE",
      "Type": "MINIMALUSER",
      "CreatedTimestamp": 1521226107.747,
      "ModifiedTimestamp": 1525297406.462,
      "Storage": {
          "StorageUtilizedInBytes": 0,
          "StorageRule": {
              "StorageAllocatedInBytes": 0,
              "StorageType": "QUOTA"
          }
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous ActivateUserà la section Référence des AWS CLI commandes.

`add-resource-permissions`

L'exemple de code suivant montre comment utiliser`add-resource-permissions`.

AWS CLI

Pour ajouter des autorisations pour une ressource

Cet exemple ajoute des autorisations à la ressource pour les principaux spécifiés.

Commande :

```
aws workdocs add-resource-permissions --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
principals Id=anonymous,Type=ANONYMOUS,Role=VIEWER
```

Sortie :

```
{
  "ShareResults": [
      {
          "PrincipalId": "anonymous",
          "Role": "VIEWER",
          "Status": "SUCCESS",
          "ShareId":
 "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
          "StatusMessage": ""
      }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous AddResourcePermissionsà la section Référence des AWS CLI commandes.

`create-comment`

L'exemple de code suivant montre comment utiliser`create-comment`.

AWS CLI

Pour ajouter un nouveau commentaire

Cet exemple ajoute un nouveau commentaire à la version de document spécifiée.

Commande :

```
aws workdocs create-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
text "This is a comment."
```

Sortie :

```
{
  "Comment": {
      "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "Text": "This is a comment.",
      "Contributor": {
          "Id": "arn:aws:iam::123456789123:user/exampleUser",
          "Username": "exampleUser",
          "GivenName": "Example",
          "Surname": "User",
          "Status": "ACTIVE"
      },
      "CreatedTimestamp": 1534799058.197,
      "Status": "PUBLISHED",
      "Visibility": "PUBLIC"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateCommentà la section Référence des AWS CLI commandes.

`create-custom-metadata`

L'exemple de code suivant montre comment utiliser`create-custom-metadata`.

AWS CLI

Pour créer des métadonnées personnalisées

Cet exemple crée des métadonnées personnalisées pour le document spécifié.

Commande :

```
aws workdocs create-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-
metadata KeyName1=example,KeyName2=example2
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous CreateCustomMetadataà la section Référence des AWS CLI commandes.

`create-folder`

L'exemple de code suivant montre comment utiliser`create-folder`.

AWS CLI

Pour créer un dossier

Cet exemple crée un dossier.

Commande :

```
aws workdocs create-folder --name documents --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Sortie :

```
{
  "Metadata": {
      "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
      "Name": "documents",
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
```

```
      "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1534450467.622,
      "ModifiedTimestamp": 1534450467.622,
      "ResourceState": "ACTIVE",
      "Signature": "",
      "Size": 0,
      "LatestVersionSize": 0
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateFolderà la section Référence des AWS CLI commandes.

`create-labels`

L'exemple de code suivant montre comment utiliser`create-labels`.

AWS CLI

Pour créer des étiquettes

Cet exemple crée une série d'étiquettes pour un document.

Commande :

```
aws workdocs create-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples" "my_documents"
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous CreateLabelsà la section Référence des AWS CLI commandes.

`create-notification-subscription`

L'exemple de code suivant montre comment utiliser`create-notification-subscription`.

AWS CLI

Pour créer un abonnement aux notifications

L'`create-notification-subscription`exemple suivant configure un abonnement aux notifications pour l' WorkDocs organisation Amazon spécifiée.

```
aws workdocs create-notification-subscription \
    --organization-id d-123456789c \
    --protocol HTTPS \
    --subscription-type ALL \
    --notification-endpoint "https://example.com/example"
```

Sortie :

```
{
    "Subscription": {
        "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
        "EndPoint": "https://example.com/example",
        "Protocol": "HTTPS"
    }
}
```

Pour plus d'informations, consultez la section S'abonner aux notifications dans le manuel Amazon WorkDocs Developer Guide.

- Pour plus de détails sur l'API, reportez-vous CreateNotificationSubscriptionà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un nouvel utilisateur

Cet exemple crée un nouvel utilisateur dans un annuaire Simple AD ou Microsoft AD.

Commande :

```
aws workdocs create-user --organization-id d-926726012c --username exampleUser2
 --email-address exampleUser2@site.awsapps.com --given-name example2Name --
surname example2Surname --password examplePa$$w0rd
```

Sortie :

```
{
  "User": {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "exampleUser2",
      "EmailAddress": "exampleUser2@site.awsapps.com",
      "GivenName": "example2Name",
      "Surname": "example2Surname",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
 "35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
      "RecycleBinFolderId":
 "9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
      "Status": "ACTIVE",
      "Type": "MINIMALUSER",
      "CreatedTimestamp": 1535478836.584,
      "ModifiedTimestamp": 1535478836.584,
      "Storage": {
          "StorageUtilizedInBytes": 0,
          "StorageRule": {
              "StorageAllocatedInBytes": 0,
              "StorageType": "QUOTA"
          }
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`deactivate-user`

L'exemple de code suivant montre comment utiliser`deactivate-user`.

AWS CLI

Pour désactiver un utilisateur

Cet exemple désactive un utilisateur actif.

Commande :

```
aws workdocs deactivate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeactivateUserà la section Référence des AWS CLI commandes.

`delete-comment`

L'exemple de code suivant montre comment utiliser`delete-comment`.

AWS CLI

Pour supprimer un commentaire spécifié d'une version de document

Cet exemple supprime le commentaire spécifié de la version de document spécifiée.

Commande :

```
aws workdocs delete-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
comment-id 1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteCommentà la section Référence des AWS CLI commandes.

`delete-custom-metadata`

L'exemple de code suivant montre comment utiliser`delete-custom-metadata`.

AWS CLI

Pour supprimer des métadonnées personnalisées d'une ressource

Cet exemple supprime toutes les métadonnées personnalisées de la ressource spécifiée.

Commande :

```
aws workdocs delete-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteCustomMetadataà la section Référence des AWS CLI commandes.

`delete-document`

L'exemple de code suivant montre comment utiliser`delete-document`.

AWS CLI

Pour supprimer un document

Cet exemple supprime le document spécifié.

Commande :

```
aws workdocs delete-document --document-
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteDocumentà la section Référence des AWS CLI commandes.

`delete-folder-contents`

L'exemple de code suivant montre comment utiliser`delete-folder-contents`.

AWS CLI

Pour supprimer le contenu d'un dossier

Cet exemple supprime le contenu du dossier spécifié.

Commande :

```
aws workdocs delete-folder-contents --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteFolderContentsà la section Référence des AWS CLI commandes.

`delete-folder`

L'exemple de code suivant montre comment utiliser`delete-folder`.

AWS CLI

Pour supprimer un dossier

Cet exemple supprime le dossier spécifié.

Commande :

```
aws workdocs delete-folder --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteFolderà la section Référence des AWS CLI commandes.

`delete-labels`

L'exemple de code suivant montre comment utiliser`delete-labels`.

AWS CLI

Pour supprimer des libellés

Cet exemple supprime les étiquettes spécifiées d'un document.

Commande :

```
aws workdocs delete-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples"
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteLabelsà la section Référence des AWS CLI commandes.

`delete-notification-subscription`

L'exemple de code suivant montre comment utiliser`delete-notification-subscription`.

AWS CLI

Pour supprimer un abonnement aux notifications

L'`delete-notification-subscription`exemple suivant supprime l'abonnement aux notifications spécifié.

```
aws workdocs delete-notification-subscription \
    --subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \
```

```
    --organization-id d-123456789c
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section S'abonner aux notifications dans le manuel Amazon WorkDocs Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteNotificationSubscriptionà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

Cet exemple supprime un utilisateur.

Commande :

```
aws workdocs delete-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`describe-activities`

L'exemple de code suivant montre comment utiliser`describe-activities`.

AWS CLI

Pour obtenir une liste des activités des utilisateurs

Cet exemple renvoie une liste des dernières activités des utilisateurs pour l'organisation spécifiée, avec une limite définie pour les deux dernières activités.

Commande :

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

Sortie :

```
{
  "UserActivities": [
      {
          "Type": "DOCUMENT_VERSION_DOWNLOADED",
          "TimeStamp": 1534800122.17,
          "Initiator": {
              "Id": "arn:aws:iam::123456789123:user/exampleUser"
          },
          "ResourceMetadata": {
              "Type": "document",
              "Name": "updatedDoc",
              "Id":
 "15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
              "Owner": {
                  "Id":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
                  "GivenName": "exampleName",
                  "Surname": "exampleSurname"
              }
          }
      },
      {
          "Type": "DOCUMENT_VERSION_VIEWED",
          "TimeStamp": 1534799079.207,
          "Initiator": {
              "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
              "GivenName": "exampleName",
              "Surname": "exampleSurname"
          },
          "ResourceMetadata": {
              "Type": "document",
              "Name": "updatedDoc",
              "Id":
 "15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
              "Owner": {
                  "Id":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
```

```
                "GivenName": "exampleName",
                "Surname": "exampleSurname"
            }
        }
    }
  ],
  "Marker":
 "DnF1ZXJ5VGhlbkZldGNoAgAAAAAAAAS7FmlTaU1OdlFTU1h1UU00VVFIbDlRWHcAAAAAAAAJTRY3bWh5eUgzaVF1ZX
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeActivitiesà la section Référence des AWS CLI commandes.

`describe-comments`

L'exemple de code suivant montre comment utiliser`describe-comments`.

AWS CLI

Pour répertorier tous les commentaires relatifs à une version de document spécifiée

Cet exemple répertorie tous les commentaires relatifs à la version de document spécifiée.

Commande :

```
aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920
```

Sortie :

```
{
  "Comments": [
      {
          "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
          "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
          "Text": "This is a comment.",
          "Contributor": {
              "Username": "arn:aws:iam::123456789123:user/exampleUser",
              "Type": "USER"
```

```
            },
            "CreatedTimestamp": 1534799058.197,
            "Status": "PUBLISHED",
            "Visibility": "PUBLIC"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeCommentsà la section Référence des AWS CLI commandes.

`describe-document-versions`

L'exemple de code suivant montre comment utiliser`describe-document-versions`.

AWS CLI

Pour récupérer les versions d'un document

Cet exemple extrait les versions du document spécifié, y compris les versions initialisées et l'URL du document source.

Commande :

```
aws workdocs describe-document-versions --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

Sortie :

```
{
    "DocumentVersions": [
        {
            "Id":
 "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
            "Name": "exampleDoc.docx",
            "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
            "Size": 13922,
            "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
            "Status": "ACTIVE",
            "CreatedTimestamp": 1534452029.587,
```

```
            "ModifiedTimestamp": 1534452029.849,
            "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
            "Source": {
                "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1534452029587-15e129dfc1875
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.docx&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
            }
        },
        {
            "Id": "1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59",
            "Name": "exampleDoc.pdf",
            "ContentType": "application/pdf",
            "Size": 425916,
            "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
            "Status": "ACTIVE",
            "CreatedTimestamp": 1529005196.082,
            "ModifiedTimestamp": 1529005196.796,
            "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
            "Source": {
                "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59?
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.pdf&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeDocumentVersionsà la section Référence des AWS CLI commandes.

`describe-folder-contents`

L'exemple de code suivant montre comment utiliser`describe-folder-contents`.

AWS CLI

Pour décrire le contenu d'un dossier

Cet exemple décrit tout le contenu actif du dossier spécifié, y compris ses documents et ses sous-dossiers, triés par date dans l'ordre croissant.

Commande :

```
aws workdocs describe-folder-contents --folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --
order ASCENDING --type ALL
```

Sortie :

```
{
  "Folders": [
      {
          "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
          "Name": "testing",
          "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
          "CreatedTimestamp": 1534450467.622,
          "ModifiedTimestamp": 1534451113.504,
          "ResourceState": "ACTIVE",
          "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
          "Size": 23019,
          "LatestVersionSize": 11537
      }
  ],
  "Documents": [
      {
          "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
          "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
          "CreatedTimestamp": 1529005196.082,
```

```
            "ModifiedTimestamp": 1534452483.01,
            "LatestVersionMetadata": {
                "Id":
 "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
                "Name": "exampleDoc.docx",
                "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
                "Size": 13922,
                "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
                "Status": "ACTIVE",
                "CreatedTimestamp": 1534452029.587,
                "ModifiedTimestamp": 1534452029.587,
                "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
            },
            "ResourceState": "ACTIVE"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeFolderContentsà la section Référence des AWS CLI commandes.

`describe-groups`

L'exemple de code suivant montre comment utiliser`describe-groups`.

AWS CLI

Pour récupérer une liste de groupes

L'`describe-groups`exemple suivant répertorie les groupes associés à l' WorkDocs organisation Amazon spécifiée.

```
aws workdocs describe-groups \
    --search-query "e" \
    --organization-id d-123456789c
```

Sortie :

```
{
    "Groups": [
```

```
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444&d-123456789c",
            "Name": "Example Group 1"
        },
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-5555&d-123456789c",
            "Name": "Example Group 2"
        }
    ]
}
```

Pour plus d'informations, consultez Getting Started with Amazon WorkDocs dans le guide d' WorkDocs administration Amazon.

- Pour plus de détails sur l'API, reportez-vous DescribeGroupsà la section Référence des AWS CLI commandes.

`describe-notification-subscriptions`

L'exemple de code suivant montre comment utiliser`describe-notification-subscriptions`.

AWS CLI

Pour récupérer la liste des abonnements aux notifications

L'`describe-notification-subscriptions`exemple suivant récupère les abonnements aux notifications pour l' WorkDocs organisation Amazon spécifiée.

```
aws workdocs describe-notification-subscriptions \
    --organization-id d-123456789c
```

Sortie :

```
{
    "Subscriptions": [
        {
            "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
            "EndPoint": "https://example.com/example",
            "Protocol": "HTTPS"
        }
    ]
}
```

Pour plus d'informations, consultez la section S'abonner aux notifications dans le manuel Amazon WorkDocs Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeNotificationSubscriptionsà la section Référence des AWS CLI commandes.

`describe-resource-permissions`

L'exemple de code suivant montre comment utiliser`describe-resource-permissions`.

AWS CLI

Pour obtenir la liste des autorisations associées à une ressource

Cet exemple renvoie une liste des autorisations pour la ressource spécifiée (document ou dossier).

Commande :

```
aws workdocs describe-resource-permissions --resource-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

Sortie :

```
{
  "Principals": [
      {
          "Id": "anonymous",
          "Type": "ANONYMOUS",
          "Roles": [
              {
                  "Role": "VIEWER",
                  "Type": "DIRECT"
              }
          ]
      },
      {
          "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "Type": "USER",
          "Roles": [
              {
                  "Role": "OWNER",
                  "Type": "DIRECT"
```

```
            }
        ]
    },
    {
        "Id": "d-926726012c",
        "Type": "ORGANIZATION",
        "Roles": [
            {
                "Role": "VIEWER",
                "Type": "INHERITED"
            }
        ]
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeResourcePermissionsà la section Référence des AWS CLI commandes.

`describe-users`

L'exemple de code suivant montre comment utiliser`describe-users`.

AWS CLI

Pour récupérer les informations relatives à des utilisateurs spécifiques

Cet exemple permet de récupérer les informations de tous les utilisateurs de l'organisation spécifiée.

Commande :

```
aws workdocs describe-users --organization-id d-926726012c
```

Sortie :

```
{
  "Users": [
    {
        "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
        "Username": "example1User",
        "OrganizationId": "d-926726012c",
```

```
            "RootFolderId":
 "3c0e3f849dd20a9771d937b9bbcc97e18796150ae56c26d64a4fa0320a2dedc9",
            "RecycleBinFolderId":
 "c277f4c4d647be1f5147b3184ffa96e1e2bf708278b696cacba68ba13b91f4fe",
            "Status": "INACTIVE",
            "Type": "USER",
            "CreatedTimestamp": 1535478999.452,
            "ModifiedTimestamp": 1535478999.452
        },
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-4444&d-926726012c",
            "Username": "example2User",
            "EmailAddress": "example2User@site.awsapps.com",
            "GivenName": "example2Name",
            "Surname": "example2Surname",
            "OrganizationId": "d-926726012c",
            "RootFolderId":
 "35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
            "RecycleBinFolderId":
 "9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
            "Status": "ACTIVE",
            "Type": "MINIMALUSER",
            "CreatedTimestamp": 1535478836.584,
            "ModifiedTimestamp": 1535478836.584
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeUsersà la section Référence des AWS CLI commandes.

`get-document-path`

L'exemple de code suivant montre comment utiliser`get-document-path`.

AWS CLI

Pour récupérer les informations relatives au chemin d'un document

Cet exemple extrait les informations de chemin (hiérarchie depuis le dossier racine) du document spécifié et inclut les noms des dossiers parents.

Commande :

```
aws workdocs get-document-path --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME
```

Sortie :

```
{
  "Path": {
      "Components": [
          {
              "Id":
 "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
              "Name": "/"
          },
          {
              "Id":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
              "Name": "Top Level Folder"
          },
          {
              "Id":
 "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
              "Name": "exampleDoc.docx"
          }
      ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDocumentPathà la section Référence des AWS CLI commandes.

`get-document-version`

L'exemple de code suivant montre comment utiliser`get-document-version`.

AWS CLI

Pour récupérer les métadonnées de version d'un document spécifié

Cet exemple extrait les métadonnées de version du document spécifié, y compris une URL source et des métadonnées personnalisées.

Commande :

```
aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata
```

Sortie :

```
{
  "Metadata": {
      "Id":
 "1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920",
      "Name": "exampleDoc",
      "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
      "Size": 11537,
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Status": "ACTIVE",
      "CreatedTimestamp": 1521672507.741,
      "ModifiedTimestamp": 1534451113.504,
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Source": {
          "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3/152167
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180820T212202Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180820%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDocumentVersionà la section Référence des AWS CLI commandes.

`get-document`

L'exemple de code suivant montre comment utiliser`get-document`.

AWS CLI

Pour récupérer les détails d'un document

Cet exemple permet de récupérer les détails du document spécifié.

Commande :

```
aws workdocs get-document --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

Sortie :

```
{
  "Metadata": {
      "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1529005196.082,
      "ModifiedTimestamp": 1534452483.01,
      "LatestVersionMetadata": {
          "Id":
 "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
          "Name": "exampleDoc.docx",
          "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
          "Size": 13922,
          "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
          "Status": "ACTIVE",
          "CreatedTimestamp": 1534452029.587,
          "ModifiedTimestamp": 1534452029.587,
          "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
      },
      "ResourceState": "ACTIVE"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetDocumentà la section Référence des AWS CLI commandes.

`get-folder-path`

L'exemple de code suivant montre comment utiliser`get-folder-path`.

AWS CLI

Pour récupérer les informations de chemin d'un dossier

Cet exemple extrait les informations de chemin (hiérarchie depuis le dossier racine) du dossier spécifié et inclut les noms des dossiers parents.

Commande :

```
aws workdocs get-folder-path --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

Sortie :

```
{
  "Path": {
      "Components": [
          {
              "Id":
 "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
              "Name": "/"
          },
          {
              "Id":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
              "Name": "Top Level Folder"
          },
          {
              "Id":
 "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
              "Name": "Sublevel Folder"
          }
      ]
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFolderPathà la section Référence des AWS CLI commandes.

`get-folder`

L'exemple de code suivant montre comment utiliser`get-folder`.

AWS CLI

Pour récupérer les métadonnées d'un dossier

Cet exemple extrait les métadonnées du dossier spécifié.

Commande :

```
aws workdocs get-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Sortie :

```
{
  "Metadata": {
      "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
      "Name": "exampleFolder",
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1534450467.622,
      "ModifiedTimestamp": 1534451113.504,
      "ResourceState": "ACTIVE",
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Size": 23019,
      "LatestVersionSize": 11537
  }
}
```

- Pour plus de détails sur l'API, reportez-vous GetFolderà la section Référence des AWS CLI commandes.

`get-resources`

L'exemple de code suivant montre comment utiliser`get-resources`.

AWS CLI

Pour récupérer des ressources partagées

L'`get-resources`exemple suivant récupère les ressources partagées avec l' WorkDocs utilisateur Amazon spécifié.

```
aws workdocs get-resources \
    --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \
    --collection-type SHARED_WITH_ME
```

Sortie :

```
{
"Folders": [],
"Documents": []
}
```

Pour plus d'informations, consultez la section Partage de fichiers et de dossiers dans le guide de WorkDocs l'utilisateur Amazon.

- Pour plus de détails sur l'API, reportez-vous GetResourcesà la section Référence des AWS CLI commandes.

`initiate-document-version-upload`

L'exemple de code suivant montre comment utiliser`initiate-document-version-upload`.

AWS CLI

Pour lancer le téléchargement d'une version de document

L'`initiate-document-version-upload`exemple suivant crée un nouvel objet de document et un nouvel objet de version.

```
aws workdocs initiate-document-version-upload \
    --name exampledocname \
    --parent-folder-
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

Sortie :

```
{
    "Metadata": {
        "Id": "feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2",
        "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
        "ParentFolderId":
 "eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189",
```

```
        "CreatedTimestamp": 1536773972.914,
        "ModifiedTimestamp": 1536773972.914,
        "LatestVersionMetadata": {
            "Id": "1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313",
            "Name": "exampledocname",
            "ContentType": "application/octet-stream",
            "Size": 0,
            "Status": "INITIALIZED",
            "CreatedTimestamp": 1536773972.914,
            "ModifiedTimestamp": 1536773972.914,
            "CreatorId": "arn:aws:iam::123456789123:user/EXAMPLE"
        },
        "ResourceState": "ACTIVE"
    },
    "UploadMetadata": {
        "UploadUrl": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2/1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313?X-Amz-
Algorithm=AWS1-ABCD-EFG234&X-Amz-Date=20180912T173932Z&X-Amz-SignedHeaders=content-
type%3Bhost%3Bx-amz-server-side-encryption&X-Amz-Expires=899&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180912%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE",
        "SignedHeaders": {
            "Content-Type": "application/octet-stream",
            "x-amz-server-side-encryption": "ABC123"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous InitiateDocumentVersionUploadà la section Référence des AWS CLI commandes.

`remove-all-resource-permissions`

L'exemple de code suivant montre comment utiliser`remove-all-resource-permissions`.

AWS CLI

Pour supprimer toutes les autorisations d'une ressource spécifiée

Cet exemple supprime toutes les autorisations de la ressource spécifiée.

Commande :

```
aws workdocs remove-all-resource-permissions --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous RemoveAllResourcePermissionsà la section Référence des AWS CLI commandes.

`remove-resource-permission`

L'exemple de code suivant montre comment utiliser`remove-resource-permission`.

AWS CLI

Pour supprimer les autorisations associées à une ressource

Cet exemple supprime les autorisations de la ressource pour le principal spécifié.

Commande :

```
aws workdocs remove-resource-permission --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-
id anonymous
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous RemoveResourcePermissionà la section Référence des AWS CLI commandes.

`update-document-version`

L'exemple de code suivant montre comment utiliser`update-document-version`.

AWS CLI

Pour changer le statut de version d'un document en Actif

Cet exemple fait passer le statut de la version du document à Active.

Commande :

```
aws workdocs update-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
version-status ACTIVE
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous UpdateDocumentVersionà la section Référence des AWS CLI commandes.

`update-document`

L'exemple de code suivant montre comment utiliser`update-document`.

AWS CLI

Pour mettre à jour un document

Cet exemple met à jour le nom et le dossier parent d'un document.

Commande :

```
aws workdocs update-document --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --
name updatedDoc --parent-folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous UpdateDocumentà la section Référence des AWS CLI commandes.

`update-folder`

L'exemple de code suivant montre comment utiliser`update-folder`.

AWS CLI

Pour mettre à jour un dossier

Cet exemple met à jour le nom et le dossier parent d'un dossier.

Commande :

```
aws workdocs update-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --
name exampleFolder1 --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Sortie :

```
None
```

- Pour plus de détails sur l'API, reportez-vous UpdateFolderà la section Référence des AWS CLI commandes.

`update-user`

L'exemple de code suivant montre comment utiliser`update-user`.

AWS CLI

Pour mettre à jour un utilisateur

Cet exemple met à jour le fuseau horaire de l'utilisateur spécifié.

Commande :

```
aws workdocs update-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-
id "America/Los_Angeles"
```

Sortie :

```
{
  "User": {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "exampleUser",
      "EmailAddress": "exampleUser@site.awsapps.com",
      "GivenName": "Example",
      "Surname": "User",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
 "c5eceb5e1a2d1d460c9d1af8330ae117fc8d39bb1d3ed6acd0992d5ff192d986",
      "RecycleBinFolderId":
 "6ca20102926ad15f04b1d248d6d6e44f2449944eda5c758f9a1e9df6a6b7fa66",
      "Status": "ACTIVE",
      "Type": "USER",
      "TimeZoneId": "America/Los_Angeles",
      "Storage": {
          "StorageUtilizedInBytes": 0,
          "StorageRule": {
              "StorageAllocatedInBytes": 53687091200,
              "StorageType": "QUOTA"
          }
      }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateUserà la section Référence des AWS CLI commandes.

WorkMail Exemples d'Amazon utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon WorkMail.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`associate-delegate-to-resource`

L'exemple de code suivant montre comment utiliser`associate-delegate-to-resource`.

AWS CLI

Pour ajouter un délégué à une ressource

La `associate-delegate-to-resource` commande suivante ajoute un délégué à une ressource.

```
aws workmail associate-delegate-to-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AssociateDelegateToResourceà la section Référence des AWS CLI commandes.

`associate-member-to-group`

L'exemple de code suivant montre comment utiliser`associate-member-to-group`.

AWS CLI

Pour ajouter un membre à un groupe

La `associate-member-to-group` commande suivante ajoute le membre spécifié à un groupe.

```
aws workmail associate-member-to-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous AssociateMemberToGroupà la section Référence des AWS CLI commandes.

`create-alias`

L'exemple de code suivant montre comment utiliser`create-alias`.

AWS CLI

Pour créer un alias

La `create-alias` commande suivante crée un alias pour l'entité spécifiée (utilisateur ou groupe).

```
aws workmail create-alias \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --alias exampleAlias@site.awsapps.com
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous CreateAliasà la section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour créer un nouveau groupe

La `create-group` commande suivante crée un nouveau groupe pour l'organisation spécifiée.

```
aws workmail create-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleGroup1
```

Sortie :

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`create-resource`

L'exemple de code suivant montre comment utiliser`create-resource`.

AWS CLI

Pour créer une nouvelle ressource

La `create-resource` commande suivante crée une nouvelle ressource (salle de réunion) pour l'organisation spécifiée.

```
aws workmail create-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleRoom1 \
    --type ROOM
```

Sortie :

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateResourceà la section Référence des AWS CLI commandes.

`create-user`

L'exemple de code suivant montre comment utiliser`create-user`.

AWS CLI

Pour créer un nouvel utilisateur

La `create-user` commande suivante crée un nouvel utilisateur.

```
aws workmail create-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleName \
    --display-name exampleDisplayName \
    --password examplePa$$w0rd
```

Sortie :

```
{
    "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"
}
```

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`delete-access-control-rule`

L'exemple de code suivant montre comment utiliser`delete-access-control-rule`.

AWS CLI

Pour supprimer une règle de contrôle d'accès

L'`delete-access-control-rule`exemple suivant supprime la règle de contrôle d'accès spécifiée de l' WorkMail organisation Amazon spécifiée.

```
aws workmail delete-access-control-rule \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --name "myRule"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec les règles de contrôle d'accès dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteAccessControlRuleà la section Référence des AWS CLI commandes.

`delete-alias`

L'exemple de code suivant montre comment utiliser`delete-alias`.

AWS CLI

Pour supprimer un alias

La `delete-alias` commande suivante supprime l'alias de l'entité spécifiée (utilisateur ou groupe).

```
aws workmail delete-alias \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --alias exampleAlias@site.awsapps.com
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteAliasà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe existant

La `delete-group` commande suivante supprime un groupe existant d'Amazon WorkMail.

```
aws workmail delete-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`delete-mailbox-permissions`

L'exemple de code suivant montre comment utiliser`delete-mailbox-permissions`.

AWS CLI

Pour supprimer les autorisations de boîte aux lettres

La `delete-mailbox-permissions` commande suivante supprime les autorisations de boîte aux lettres précédemment accordées à un utilisateur ou à un groupe. L'entité représente l'utilisateur propriétaire de la boîte aux lettres et le bénéficiaire représente l'utilisateur ou le groupe pour lequel les autorisations doivent être supprimées.

```
aws workmail delete-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --grantee-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteMailboxPermissionsà la section Référence des AWS CLI commandes.

`delete-resource`

L'exemple de code suivant montre comment utiliser`delete-resource`.

AWS CLI

Pour supprimer une ressource existante

La `delete-resource` commande suivante supprime une ressource existante d'Amazon WorkMail.

```
aws workmail delete-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
```

```
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteResourceà la section Référence des AWS CLI commandes.

`delete-user`

L'exemple de code suivant montre comment utiliser`delete-user`.

AWS CLI

Pour supprimer un utilisateur

La `delete-user` commande suivante supprime l'utilisateur spécifié d'Amazon WorkMail et de tous les systèmes suivants.

```
aws workmail delete-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`deregister-from-work-mail`

L'exemple de code suivant montre comment utiliser`deregister-from-work-mail`.

AWS CLI

Pour désactiver une entité existante

La `deregister-from-work-mail` commande suivante empêche une entité existante (utilisateur, groupe ou ressource) d'utiliser Amazon WorkMail.

```
aws workmail deregister-from-work-mail \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
```

```
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeregisterFromWorkMailà la section Référence des AWS CLI commandes.

`describe-group`

L'exemple de code suivant montre comment utiliser`describe-group`.

AWS CLI

Pour récupérer des informations pour un groupe

La `describe-group` commande suivante permet de récupérer des informations sur le groupe spécifié.

```
aws workmail describe-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Sortie :

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
    "Name": "exampleGroup1",
    "State": "ENABLED"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeGroupà la section Référence des AWS CLI commandes.

`describe-organization`

L'exemple de code suivant montre comment utiliser`describe-organization`.

AWS CLI

Pour récupérer des informations pour une organisation

La `describe-organization` commande suivante permet de récupérer les informations relatives à l' WorkMail organisation Amazon spécifiée.

```
aws workmail describe-organization \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Sortie :

```
{
    "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
    "Alias": "alias",
    "State": "Active",
    "DirectoryId": "d-926726012c",
    "DirectoryType": "VpcDirectory",
    "DefaultMailDomain": "site.awsapps.com",
    "CompletedDate": 1522693605.468,
    "ARN": "arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza"
}
```

Pour plus d'informations, consultez Working with Organizations dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous DescribeOrganizationà la section Référence des AWS CLI commandes.

`describe-resource`

L'exemple de code suivant montre comment utiliser`describe-resource`.

AWS CLI

Pour récupérer les informations relatives à une ressource

La `describe-resource` commande suivante permet de récupérer des informations sur la ressource spécifiée.

```
aws workmail describe-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

Sortie :

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
    "Name": "exampleRoom1",
    "Type": "ROOM",
    "BookingOptions": {
        "AutoAcceptRequests": true,
        "AutoDeclineRecurringRequests": false,
        "AutoDeclineConflictingRequests": true
    },
    "State": "ENABLED"
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeResourceà la section Référence des AWS CLI commandes.

`describe-user`

L'exemple de code suivant montre comment utiliser`describe-user`.

AWS CLI

Pour récupérer les informations utilisateur

La `describe-user` commande suivante permet de récupérer des informations sur l'utilisateur spécifié.

```
aws workmail describe-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Sortie :

```
{
    "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",
    "Name": "exampleUser1",
    "Email": "exampleUser1@site.awsapps.com",
    "DisplayName": "",
    "State": "ENABLED",
    "UserRole": "USER",
    "EnabledDate": 1532459261.827
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeUserà la section Référence des AWS CLI commandes.

`disassociate-delegate-from-resource`

L'exemple de code suivant montre comment utiliser`disassociate-delegate-from-resource`.

AWS CLI

Pour supprimer un membre d'une ressource

La `disassociate-delegate-from-resource` commande suivante supprime le membre spécifié d'une ressource.

```
ws workmail disassociate-delegate-from-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DisassociateDelegateFromResourceà la section Référence des AWS CLI commandes.

`disassociate-member-from-group`

L'exemple de code suivant montre comment utiliser`disassociate-member-from-group`.

AWS CLI

Pour supprimer un membre d'un groupe

La `disassociate-member-from-group` commande suivante supprime le membre spécifié d'un groupe.

```
aws workmail disassociate-member-from-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DisassociateMemberFromGroupà la section Référence des AWS CLI commandes.

`get-access-control-effect`

L'exemple de code suivant montre comment utiliser`get-access-control-effect`.

AWS CLI

Pour obtenir l'effet des règles de contrôle d'accès

L'`get-access-control-effect`exemple suivant montre l'effet des règles de contrôle d'accès de WorkMail l'organisation Amazon spécifiée pour l'adresse IP, l'action du protocole d'accès et l'ID utilisateur spécifiés.

```
aws workmail get-access-control-effect \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --ip-address "192.0.2.0" \
    --action "WindowsOutlook" \
    --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333"
```

Sortie :

```
{
    "Effect": "DENY",
    "MatchedRules": [
        "myRule"
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec les règles de contrôle d'accès dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous GetAccessControlEffectà la section Référence des AWS CLI commandes.

`get-mailbox-details`

L'exemple de code suivant montre comment utiliser`get-mailbox-details`.

AWS CLI

Pour obtenir les détails de la boîte aux lettres d'un utilisateur

La `get-mailbox-details` commande suivante permet de récupérer des informations sur la boîte aux lettres de l'utilisateur spécifié.

```
aws workmail get-mailbox-details \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Sortie :

```
{
    "MailboxQuota": 51200,
    "MailboxSize": 0.03890800476074219
}
```

Pour plus d'informations, consultez la section Gestion des comptes utilisateurs dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous GetMailboxDetailsà la section Référence des AWS CLI commandes.

`list-access-control-rules`

L'exemple de code suivant montre comment utiliser`list-access-control-rules`.

AWS CLI

Pour répertorier les règles de contrôle d'accès

L'`list-access-control-rules`exemple suivant répertorie les règles de contrôle d'accès pour l' WorkMail organisation Amazon spécifiée.

```
aws workmail list-access-control-rules \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

Sortie :

```
{
    "Rules": [
```

```
        {
            "Name": "default",
            "Effect": "ALLOW",
            "Description": "Default WorkMail Rule",
            "DateCreated": 0.0,
            "DateModified": 0.0
        },
        {
            "Name": "myRule",
            "Effect": "DENY",
            "Description": "my rule",
            "UserIds": [
            "S-1-1-11-1111111111-2222222222-3333333333-3333"
            ],
            "DateCreated": 1581635628.0,
            "DateModified": 1581635628.0
        }
    ]
}
```

Pour plus d'informations, consultez la section Travailler avec les règles de contrôle d'accès dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous ListAccessControlRulesà la section Référence des AWS CLI commandes.

`list-aliases`

L'exemple de code suivant montre comment utiliser`list-aliases`.

AWS CLI

Pour répertorier les alias d'un membre

La `list-aliases` commande suivante répertorie les alias du membre spécifié (utilisateur ou groupe).

```
aws workmail list-aliases \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Sortie :

```
{
    "Aliases": [
        "exampleAlias@site.awsapps.com",
        "exampleAlias1@site.awsapps.com"
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListAliasesà la section Référence des AWS CLI commandes.

`list-group-members`

L'exemple de code suivant montre comment utiliser`list-group-members`.

AWS CLI

Pour répertorier les membres du groupe

La `list-group-members` commande suivante répertorie les membres du groupe spécifié.

```
aws workmail list-group-members \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Sortie :

```
{
    "Members": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Name": "exampleUser1",
            "Type": "USER",
            "State": "ENABLED",
            "EnabledDate": 1532459261.827
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGroupMembersà la section Référence des AWS CLI commandes.

`list-groups`

L'exemple de code suivant montre comment utiliser`list-groups`.

AWS CLI

Pour récupérer une liste de groupes

La `list-groups` commande suivante permet de récupérer les résumés des groupes de l'organisation spécifiée.

```
aws workmail list-groups \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Sortie :

```
{
    "Groups": [
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "Name": "exampleGroup1",
            "State": "DISABLED"
        },
        {
            "Id": "S-4-4-44-1122222222-2222233333-3333334444-4444",
            "Name": "exampleGroup2",
            "State": "ENABLED"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListGroupsà la section Référence des AWS CLI commandes.

`list-mailbox-permissions`

L'exemple de code suivant montre comment utiliser`list-mailbox-permissions`.

AWS CLI

Pour récupérer les autorisations de boîte aux lettres

La `list-mailbox-permissions` commande suivante récupère les autorisations de boîte aux lettres associées à la boîte aux lettres de l'entité spécifiée.

```
aws workmail list-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Sortie :

```
{
    "Permissions": [
        {
            "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "GranteeType": "USER",
            "PermissionValues": [
                "FULL_ACCESS"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListMailboxPermissionsà la section Référence des AWS CLI commandes.

`list-organizations`

L'exemple de code suivant montre comment utiliser`list-organizations`.

AWS CLI

Pour récupérer une liste d'organisations

La `list-organizations` commande suivante permet de récupérer des résumés des organisations du client.

```
aws workmail list-organizations
```

Sortie :

```
{
```

```
    "OrganizationSummaries": [
        {
            "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
            "Alias": "exampleAlias",
            "State": "Active"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListOrganizationsà la section Référence des AWS CLI commandes.

`list-resource-delegates`

L'exemple de code suivant montre comment utiliser`list-resource-delegates`.

AWS CLI

Pour répertorier les délégués d'une ressource

La `list-resource-delegates` commande suivante permet de récupérer les délégués associés à la ressource spécifiée.

```
aws workmail list-resource-delegates \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

Sortie :

```
{
    "Delegates": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Type": "USER"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourceDelegatesà la section Référence des AWS CLI commandes.

`list-resources`

L'exemple de code suivant montre comment utiliser`list-resources`.

AWS CLI

Pour récupérer une liste de ressources

La `list-resources` commande suivante permet de récupérer des résumés des ressources pour l'organisation spécifiée.

```
aws workmail list-resources \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Sortie :

```
{
    "Resources": [
        {
            "Id": "r-7afe0efbade843a58cdc10251fce992c",
            "Name": "exampleRoom1",
            "Type": "ROOM",
            "State": "ENABLED"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListResourcesà la section Référence des AWS CLI commandes.

`list-tags-for-resource`

L'exemple de code suivant montre comment utiliser`list-tags-for-resource`.

AWS CLI

Pour répertorier les balises d'une ressource

L'`list-tags-for-resource`exemple suivant répertorie les balises pour l' WorkMail organisation Amazon spécifiée.

```
aws workmail list-tags-for-resource \
```

```
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza
```

Sortie :

```
{
    "Tags": [
        {
            "Key": "priority",
            "Value": "1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Marquage d'une organisation dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous ListTagsForResourceà la section Référence des AWS CLI commandes.

`list-users`

L'exemple de code suivant montre comment utiliser`list-users`.

AWS CLI

Pour récupérer une liste d'utilisateurs

La `list-users` commande suivante permet de récupérer les résumés des utilisateurs de l'organisation spécifiée.

```
aws workmail list-users \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Sortie :

```
{
    "Users": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Email": "exampleUser1@site.awsapps.com",
```

```
            "Name": "exampleUser1",
            "State": "ENABLED",
            "UserRole": "USER",
            "EnabledDate": 1532459261.827
        },
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "Name": "exampleGuestUser",
            "State": "DISABLED",
            "UserRole": "SYSTEM_USER"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`put-access-control-rule`

L'exemple de code suivant montre comment utiliser`put-access-control-rule`.

AWS CLI

Pour mettre une nouvelle règle de contrôle d'accès

L'`put-access-control-rule`exemple suivant refuse à l'utilisateur spécifié l'accès à l' WorkMail organisation Amazon spécifiée.

```
aws workmail put-access-control-rule \
    --name "myRule" \
    --effect "DENY" \
    --description "my rule" \
    --user-ids "S-1-1-11-1111111111-2222222222-3333333333-3333" \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Travailler avec les règles de contrôle d'accès dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous PutAccessControlRuleà la section Référence des AWS CLI commandes.

`put-mailbox-permissions`

L'exemple de code suivant montre comment utiliser`put-mailbox-permissions`.

AWS CLI

Pour définir les autorisations de boîte aux lettres

La `put-mailbox-permissions` commande suivante définit les autorisations d'accès complètes pour le bénéficiaire spécifié (utilisateur ou groupe). L'entité représente le propriétaire de la boîte aux lettres.

```
aws workmail put-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --grantee-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --permission-values FULL_ACCESS
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutMailboxPermissionsà la section Référence des AWS CLI commandes.

`register-to-work-mail`

L'exemple de code suivant montre comment utiliser`register-to-work-mail`.

AWS CLI

Pour enregistrer une entité existante ou désactivée

La `register-to-work-mail` commande suivante permet à l'entité existante spécifiée (utilisateur, groupe ou ressource) d'utiliser Amazon WorkMail.

```
aws workmail register-to-work-mail \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --email exampleGroup1@site.awsapps.com
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RegisterToWorkMailà la section Référence des AWS CLI commandes.

`reset-password`

L'exemple de code suivant montre comment utiliser`reset-password`.

AWS CLI

Pour réinitialiser le mot de passe d'un utilisateur

La `reset-password` commande suivante réinitialise le mot de passe de l'utilisateur spécifié.

```
aws workmail reset-password \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --password examplePa$$w0rd
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous ResetPasswordà la section Référence des AWS CLI commandes.

`tag-resource`

L'exemple de code suivant montre comment utiliser`tag-resource`.

AWS CLI

Pour appliquer un tag à une ressource

L'`tag-resource`exemple suivant applique une balise avec la clé « priority » et la valeur « 1 » à l'WorkMail organisation Amazon spécifiée.

```
aws workmail tag-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza \
    --tags "Key=priority,Value=1"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage d'une organisation dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous TagResourceà la section Référence des AWS CLI commandes.

`untag-resource`

L'exemple de code suivant montre comment utiliser`untag-resource`.

AWS CLI

Pour annuler le balisage d'une ressource

L'`untag-resource`exemple suivant supprime le tag spécifié de l' WorkMail organisation Amazon spécifiée.

```
aws workmail untag-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-n1pq2345678r901st2u3vx45x6789yza \
    --tag-keys "priority"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Marquage d'une organisation dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous UntagResourceà la section Référence des AWS CLI commandes.

`update-mailbox-quota`

L'exemple de code suivant montre comment utiliser`update-mailbox-quota`.

AWS CLI

Pour mettre à jour le quota de boîte aux lettres d'un utilisateur

La `update-mailbox-quota` commande suivante modifie le quota de boîte aux lettres de l'utilisateur spécifié.

```
aws workmail update-mailbox-quota \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --mailbox-quota 40000
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des comptes utilisateurs dans le manuel Amazon WorkMail Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous UpdateMailboxQuotaà la section Référence des AWS CLI commandes.

`update-primary-email-address`

L'exemple de code suivant montre comment utiliser`update-primary-email-address`.

AWS CLI

Pour mettre à jour une adresse e-mail principale

La `update-primary-email-address` commande suivante met à jour l'adresse e-mail principale de l'entité spécifiée (utilisateur, groupe ou ressource).

```
aws workmail update-primary-email-address \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --email exampleUser2@site.awsapps.com
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdatePrimaryEmailAddressà la section Référence des AWS CLI commandes.

`update-resource`

L'exemple de code suivant montre comment utiliser`update-resource`.

AWS CLI

Pour mettre à jour une ressource

La `update-resource` commande suivante met à jour le nom de la ressource spécifiée.

```
aws workmail update-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c \
```

```
    --name exampleRoom2
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous UpdateResourceà la section Référence des AWS CLI commandes.

Exemples d'Amazon WorkMail Message Flow utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l' AWS Command Line Interface aide d'Amazon WorkMail Message Flow.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`get-raw-message-content`

L'exemple de code suivant montre comment utiliser`get-raw-message-content`.

AWS CLI

Pour obtenir le contenu brut d'un e-mail

L'`get-raw-message-content`exemple suivant récupère le contenu brut d'un e-mail en transit et l'envoie dans un fichier texte nommé`test`.

```
aws workmailmessageflow get-raw-message-content \
    --message-id a1b2cd34-ef5g-6h7j-kl8m-npq9012345rs \
    test
```

Contenu du fichier `test` après l'exécution de la commande :

```
Subject: Hello World
From: =?UTF-8?Q?marymajor_marymajor?= <marymajor@example.com>
To: =?UTF-8?Q?mateojackson=40example=2Enet?= <mateojackson@example.net>
Date: Thu, 7 Nov 2019 19:22:46 +0000
Mime-Version: 1.0
Content-Type: multipart/alternative;
 boundary="=_EXAMPLE+"
References: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>
X-Priority: 3 (Normal)
X-Mailer: Amazon WorkMail
Thread-Index: EXAMPLE
Thread-Topic: Hello World
Message-Id: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>

This is a multi-part message in MIME format. Your mail reader does not
understand MIME message format.
--=_EXAMPLE+
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

hello world


--=_EXAMPLE+
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML><html>
<head>
<meta name=3D"Generator" content=3D"Amazon WorkMail v3.0-4510">
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Dutf-8">=

<title>testing</title>
</head>
<body>
<p style=3D"margin: 0px; font-family: Arial, Tahoma, Helvetica, sans-seri=
f; font-size: small;">hello world</p>
</body>
</html>
--=_EXAMPLE+--
```

Pour plus d'informations, consultez la section Récupération du contenu des messages avec AWS Lambda dans le manuel WorkMail Amazon Administrator Guide.

- Pour plus de détails sur l'API, reportez-vous GetRawMessageContentà la section Référence des AWS CLI commandes.

WorkSpaces exemples utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du AWS Command Line Interface with WorkSpaces.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`create-tags`

L'exemple de code suivant montre comment utiliser`create-tags`.

AWS CLI

Pour ajouter des balises à un WorkSpace

L'`create-tags`exemple suivant ajoute les balises spécifiées aux balises spécifiées WorkSpace.

```
aws workspaces create-tags \
    --resource-id ws-dk1xzr417 \
    --tags Key=Department,Value=Finance
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section WorkSpaces Ressources relatives aux balises dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateTagsà la section Référence des AWS CLI commandes.

`create-workspaces`

L'exemple de code suivant montre comment utiliser`create-workspaces`.

AWS CLI

Exemple 1 : pour créer un AlwaysOn WorkSpace

L'`create-workspaces`exemple suivant crée un AlwaysOn WorkSpace pour l'utilisateur spécifié, en utilisant le répertoire et le bundle spécifiés.

```
aws workspaces create-workspaces \
    --workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

Sortie :

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-kcqms853t",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mateo",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc"
        }
    ]
}
```

Exemple 2 : pour créer un AutoStop WorkSpace

L'`create-workspaces`exemple suivant crée un AutoStop WorkSpace pour l'utilisateur spécifié, en utilisant le répertoire et le bundle spécifiés.

```
aws workspaces create-workspaces \
    --
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties
```

Sortie :

```
{
    "FailedRequests": [],
```

```
    "PendingRequests": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mary",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc"
        }
    ]
}
```

Exemple 3 : pour créer un système découplé par l'utilisateur WorkSpace

L'`create-workspaces`exemple suivant crée un découplage utilisateur WorkSpace en définissant le nom d'utilisateur sur et en spécifiant un WorkSpace nom`[UNDEFINED]`, un ID de répertoire et un ID de bundle.

```
aws workspaces create-workspaces \
    --workspaces
 DirectoryId=d-926722edaf,UserName='"[UNDEFINED]"',WorkspaceName=MaryWorkspace1,BundleId=wsb
```

Sortie :

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-abcd1234",
            "DirectoryId": "d-926722edaf",
            "UserName": "[UNDEFINED]",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc",
            "WorkspaceName": "MaryWorkspace1"
        }
    ]
}
```

Pour plus d'informations, consultez la section Lancer un bureau virtuel dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, reportez-vous CreateWorkspacesà la section Référence des AWS CLI commandes.

`delete-tags`

L'exemple de code suivant montre comment utiliser`delete-tags`.

AWS CLI

Pour supprimer un tag d'un WorkSpace

L'`delete-tags`exemple suivant supprime la balise spécifiée de la balise spécifiée WorkSpace.

```
aws workspaces delete-tags \
    --resource-id ws-dk1xzr417 \
    --tag-keys Department
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section WorkSpaces Ressources relatives aux balises dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, reportez-vous DeleteTagsà la section Référence des AWS CLI commandes.

`deregister-workspace-directory`

L'exemple de code suivant montre comment utiliser`deregister-workspace-directory`.

AWS CLI

Pour désenregistrer un annuaire

L'`deregister-workspace-directory`exemple suivant annule l'enregistrement du répertoire spécifié.

```
aws workspaces deregister-workspace-directory \
    --directory-id d-926722edaf
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Enregistrer un annuaire WorkSpaces dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, reportez-vous DeregisterWorkspaceDirectoryà la section Référence des AWS CLI commandes.

`describe-tags`

L'exemple de code suivant montre comment utiliser`describe-tags`.

AWS CLI

Pour décrire les balises d'un WorkSpace

L'`describe-tags`exemple suivant décrit les balises pour le paramètre spécifié WorkSpace.

```
aws workspaces describe-tags \
    --resource-id ws-dk1xzr417
```

Sortie :

```
{
    "TagList": [
        {
            "Key": "Department",
            "Value": "Finance"
        }
    ]
}
```

Pour plus d'informations, consultez la section WorkSpaces Ressources relatives aux balises dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, voir DescribeTagsla section Référence des AWS CLI commandes.

`describe-workspace-bundles`

L'exemple de code suivant montre comment utiliser`describe-workspace-bundles`.

AWS CLI

Pour répertorier les offres groupées proposées par Amazon

L'`describe-workspace-bundles`exemple suivant répertorie les noms et IDs les offres groupées fournis par Amazon, sous forme de tableau et triés par nom.

```
aws workspaces describe-workspace-bundles \
```

```
    --owner AMAZON \
    --query "Bundles[*].[Name, BundleId]"
```

Sortie :

```
[
    [
        "Standard with Amazon Linux 2",
        "wsb-clj85qzj1"
    ],
    [
        "Performance with Windows 10 (Server 2016 based)",
        "wsb-gm4d5tx2v"
    ],
    [
        "PowerPro with Windows 7",
        "wsb-1pzkp0bx4"
    ],
    [
        "Power with Amazon Linux 2",
        "wsb-2bs6k5lgn"
    ],
    [
        "Graphics with Windows 10 (Server 2019 based)",
        "wsb-03gyjnfyy"
    ],
    ...
]
```

Pour plus d'informations, consultez WorkSpaces les offres groupées et les images dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir DescribeWorkspaceBundlesla section Référence des AWS CLI commandes.

`describe-workspace-directories`

L'exemple de code suivant montre comment utiliser`describe-workspace-directories`.

AWS CLI

Pour décrire un annuaire enregistré

L'`describe-workspace-directories`exemple suivant décrit le répertoire enregistré spécifié.

```
aws workspaces describe-workspace-directories \
    --directory-ids d-926722edaf
```

Sortie :

```
{
    "Directories": [
        {
            "DirectoryId": "d-926722edaf",
            "Alias": "d-926722edaf",
            "DirectoryName": "example.com",
            "RegistrationCode": "WSpdx+9RJ8JT",
            "SubnetIds": [
                "subnet-9d19c4c6",
                "subnet-500d5819"
            ],
            "DnsIpAddresses": [
                "172.16.1.140",
                "172.16.0.30"
            ],
            "CustomerUserName": "Administrator",
            "IamRoleId": "arn:aws:iam::123456789012:role/workspaces_DefaultRole",
            "DirectoryType": "SIMPLE_AD",
            "WorkspaceSecurityGroupId": "sg-0d89e927e5645d7c5",
            "State": "REGISTERED",
            "WorkspaceCreationProperties": {
                "EnableInternetAccess": false,
                "UserEnabledAsLocalAdministrator": true,
                "EnableMaintenanceMode": true
            },
            "WorkspaceAccessProperties": {
                "DeviceTypeWindows": "ALLOW",
                "DeviceTypeOsx": "ALLOW",
                "DeviceTypeWeb": "DENY",
                "DeviceTypeIos": "ALLOW",
                "DeviceTypeAndroid": "ALLOW",
                "DeviceTypeChromeOs": "ALLOW",
                "DeviceTypeZeroClient": "ALLOW",
                "DeviceTypeLinux": "DENY"
            },
            "Tenancy": "SHARED",
```

```
            "SelfservicePermissions": {
                "RestartWorkspace": "ENABLED",
                "IncreaseVolumeSize": "DISABLED",
                "ChangeComputeType": "DISABLED",
                "SwitchRunningMode": "DISABLED",
                "RebuildWorkspace": "DISABLED"
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Gérer les annuaires pour les WorkSpaces particuliers dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, voir DescribeWorkspaceDirectoriesla section Référence des AWS CLI commandes.

`describe-workspaces-connection-status`

L'exemple de code suivant montre comment utiliser`describe-workspaces-connection-status`.

AWS CLI

Pour décrire l'état de connexion d'un WorkSpace

L'`describe-workspaces-connection-status`exemple suivant décrit l'état de connexion du paramètre spécifié WorkSpace.

```
aws workspaces describe-workspaces-connection-status \
    --workspace-ids ws-dk1xzr417
```

Sortie :

```
{
    "WorkspacesConnectionStatus": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "ConnectionState": "CONNECTED",
            "ConnectionStateCheckTimestamp": 1662526214.744
        }
    ]
```

```
}
```

Pour plus d'informations, consultez la section Administrer votre WorkSpaces dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, voir DescribeWorkspacesConnectionStatusla section Référence des AWS CLI commandes.

`describe-workspaces`

L'exemple de code suivant montre comment utiliser`describe-workspaces`.

AWS CLI

Pour décrire un WorkSpace

L'`describe-workspaces`exemple suivant décrit ce qui est spécifié WorkSpace.

```
aws workspaces describe-workspaces \
    --workspace-ids ws-dk1xzr417
```

Sortie :

```
{
    "Workspaces": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mary",
            "IpAddress": "172.16.0.175",
            "State": "STOPPED",
            "BundleId": "wsb-0zsvgp8fc",
            "SubnetId": "subnet-500d5819",
            "ComputerName": "WSAMZN-RBSLTTD9",
            "WorkspaceProperties": {
                "RunningMode": "AUTO_STOP",
                "RunningModeAutoStopTimeoutInMinutes": 60,
                "RootVolumeSizeGib": 80,
                "UserVolumeSizeGib": 10,
                "ComputeTypeName": "VALUE"
            },
            "ModificationStates": []
```

```
        }
    ]
}
```

Pour plus d'informations, consultez la section Administrer votre WorkSpaces dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, voir DescribeWorkspacesla section Référence des AWS CLI commandes.

`migrate-workspace`

L'exemple de code suivant montre comment utiliser`migrate-workspace`.

AWS CLI

Pour migrer un WorkSpace

L'`migrate-workspace`exemple suivant fait migrer le bundle spécifié WorkSpace vers le bundle spécifié.

```
aws workspaces migrate-workspace \
    --source-workspace-id ws-dk1xzr417 \
    --bundle-id wsb-j4dky1gs4
```

Sortie :

```
{
    "SourceWorkspaceId": "ws-dk1xzr417",
    "TargetWorkspaceId": "ws-x5h1lbkp5"
}
```

Pour plus d'informations, consultez Migrate a WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir MigrateWorkspacela section Référence des AWS CLI commandes.

`modify-workspace-creation-properties`

L'exemple de code suivant montre comment utiliser`modify-workspace-creation-properties`.

AWS CLI

Pour modifier une propriété de WorkSpace création d'un répertoire

L'`modify-workspace-creation-properties`exemple suivant active la `EnableInternetAccess` propriété pour le répertoire spécifié. Cela permet l'attribution automatique d'adresses IP publiques pour les adresses IP WorkSpaces créées pour l'annuaire.

```
aws workspaces modify-workspace-creation-properties \
    --resource-id d-926722edaf \
    --workspace-creation-properties EnableInternetAccess=true
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Mettre à jour les informations de votre répertoire WorkSpaces dans le guide WorkSpaces d'administration Amazon.

- Pour plus de détails sur l'API, voir ModifyWorkspaceCreationPropertiesla section Référence des AWS CLI commandes.

`modify-workspace-properties`

L'exemple de code suivant montre comment utiliser`modify-workspace-properties`.

AWS CLI

Pour modifier le mode de fonctionnement d'un WorkSpace

L'`modify-workspace-properties`exemple suivant définit le mode d'exécution du paramètre spécifié WorkSpace sur`AUTO_STOP`.

```
aws workspaces modify-workspace-properties \
    --workspace-id ws-dk1xzr417 \
    --workspace-properties RunningMode=AUTO_STOP
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Modifier un WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir ModifyWorkspacePropertiesla section Référence des AWS CLI commandes.

`modify-workspace-state`

L'exemple de code suivant montre comment utiliser`modify-workspace-state`.

AWS CLI

Pour modifier l'état d'un WorkSpace

L'`modify-workspace-state`exemple suivant définit l'état de la valeur spécifiée WorkSpace sur`ADMIN_MAINTENANCE`.

```
aws workspaces modify-workspace-state \
    --workspace-id ws-dk1xzr417 \
    --workspace-state ADMIN_MAINTENANCE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section WorkSpace maintenance dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir ModifyWorkspaceStatela section Référence des AWS CLI commandes.

`reboot-workspaces`

L'exemple de code suivant montre comment utiliser`reboot-workspaces`.

AWS CLI

Pour redémarrer un WorkSpace

L'`reboot-workspaces`exemple suivant redémarre le fichier spécifié WorkSpace.

```
aws workspaces reboot-workspaces \
    --reboot-workspace-requests ws-dk1xzr417
```

Sortie :

```
{
    "FailedRequests": []
```

```
}
```

Pour plus d'informations, consultez Reboot a WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir RebootWorkspacesla section Référence des AWS CLI commandes.

`rebuild-workspaces`

L'exemple de code suivant montre comment utiliser`rebuild-workspaces`.

AWS CLI

Pour reconstruire un WorkSpace

L'`rebuild-workspaces`exemple suivant reconstruit le fichier spécifié WorkSpace.

```
aws workspaces rebuild-workspaces \
    --rebuild-workspace-requests ws-dk1xzr417
```

Sortie :

```
{
    "FailedRequests": []
}
```

Pour plus d'informations, consultez Rebuild a WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir RebuildWorkspacesla section Référence des AWS CLI commandes.

`register-workspace-directory`

L'exemple de code suivant montre comment utiliser`register-workspace-directory`.

AWS CLI

Pour enregistrer un annuaire

L'`register-workspace-directory`exemple suivant enregistre le répertoire spécifié pour une utilisation avec Amazon WorkSpaces.

```
aws workspaces register-workspace-directory \
    --directory-id d-926722edaf
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Enregistrer un AWS annuaire Directory Service existant auprès de WorkSpaces Personal dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir RegisterWorkspaceDirectorylla section Référence des AWS CLI commandes.

`restore-workspace`

L'exemple de code suivant montre comment utiliser`restore-workspace`.

AWS CLI

Pour restaurer un WorkSpace

L'`restore-workspace`exemple suivant restaure le paramètre spécifié WorkSpace.

```
aws workspaces restore-workspace \
    --workspace-id ws-dk1xzr417
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Restore a WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir RestoreWorkspacela section Référence des AWS CLI commandes.

`start-workspaces`

L'exemple de code suivant montre comment utiliser`start-workspaces`.

AWS CLI

Pour démarrer un AutoStop WorkSpace

L'`start-workspaces`exemple suivant démarre le fichier spécifié WorkSpace. Le mode de fonctionnement WorkSpace doit être de`AutoStop`.

```
aws workspaces start-workspaces \
    --start-workspace-requests WorkspaceId=ws-dk1xzr417
```

Sortie :

```
{
    "FailedRequests": []
}
```

Pour plus d'informations, consultez la section Stop and start an AutoStop WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir StartWorkspacesla section Référence des AWS CLI commandes.

`stop-workspaces`

L'exemple de code suivant montre comment utiliser`stop-workspaces`.

AWS CLI

Pour arrêter un AutoStop WorkSpace

L'`stop-workspaces`exemple suivant arrête le paramètre spécifié WorkSpace. Le mode de fonctionnement WorkSpace doit être de`AutoStop`.

```
aws workspaces stop-workspaces \
    --stop-workspace-requests WorkspaceId=ws-dk1xzr417
```

Sortie :

```
{
    "FailedRequests": []
}
```

Pour plus d'informations, consultez la section Stop and start an AutoStop WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir StopWorkspacesla section Référence des AWS CLI commandes.

`terminate-workspaces`

L'exemple de code suivant montre comment utiliser`terminate-workspaces`.

AWS CLI

Pour mettre fin à un WorkSpace

L'`terminate-workspaces`exemple suivant met fin à l'espace de travail spécifié.

```
aws workspaces terminate-workspaces \
    --terminate-workspace-requests ws-dk1xzr417
```

Sortie :

```
{
    "FailedRequests": []
}
```

Pour plus d'informations, consultez Supprimer un WorkSpace dans le guide d' WorkSpaces administration Amazon.

- Pour plus de détails sur l'API, voir TerminateWorkspacesla section Référence des AWS CLI commandes.

Exemples de X-Ray utilisant AWS CLI

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide de AWS Command Line Interface with X-Ray.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`batch-traces-get`

L'exemple de code suivant montre comment utiliser`batch-traces-get`.

AWS CLI

Pour obtenir une liste de traces

L'`batch-get-traces`exemple suivant récupère une liste de traces spécifiées par un ID. Le suivi complet inclut un document pour chaque segment, compilé à partir de l'ensemble des documents de segments reçus pour le même ID de suivi.

```
aws xray batch-get-traces \
    --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

Sortie :

```
{
    "Traces": [
        {
            "Id": "1-5d82881a-0a9126e92a73e971eed891b9",
            "Duration": 0.232,
            "Segments": [
                {
                    "Id": "54aff5735b12dd28",
                    "Document": "{\"id\":\"54aff5735b12dd28\",\"name\":
\"Scorekeep\",\"start_time\":1.568835610432E9,\"end_time\":1.568835610664E9,
\"http\":{\"request\":{\"url\":\"http://scorekeep-env-1.m4fg2pfzpv.us-
east-2.elasticbeanstalk.com/api/user\",\"method\":\"POST\",\"user_agent\":
\"curl/7.59.0\",\"client_ip\":\"52.95.4.28\",\"x_forwarded_for\":true},
\"response\":{\"status\":200}},\"aws\":{\"elastic_beanstalk\":{\"version_label
\":\"Sample Application-1\",\"deployment_id\":3,\"environment_name\":\"Scorekeep-
env-1\"},\"ec2\":{\"availability_zone\":\"us-east-2b\",\"instance_id\":
\"i-0e3cf4d2de0f3f37a\"},\"xray\":{\"sdk_version\":\"1.1.0\",\"sdk\":\"X-Ray for
 Java\"}},\"service\":{\"runtime\":\"OpenJDK 64-Bit Server VM\",\"runtime_version
\":\"1.8.0_222\"},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",
\"origin\":\"AWS::ElasticBeanstalk::Environment\",\"subsegments\":[{\"id\":
```

```
\"2d6900034ccfe558\",\"name\":\"DynamoDB\",\"start_time\":1.568835610658E9,
\"end_time\":1.568835610664E9,\"http\":{\"response\":{\"status\":200,
\"content_length\":61}},\"aws\":{\"table_name\":\"scorekeep-user\",\"operation\":
\"UpdateItem\",\"request_id\":\"TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG
\",\"resource_names\":[\"scorekeep-user\"]},\"namespace\":\"aws\"}]}"
                },
                {
                    "Id": "0f278b6334c34e6b",
                    "Document": "{\"id\":\"0f278b6334c34e6b\",\"name\":
\"DynamoDB\",\"start_time\":1.568835610658E9,\"end_time\":1.568835610664E9,
\"parent_id\":\"2d6900034ccfe558\",\"inferred\":true,\"http\":{\"response
\":{\"status\":200,\"content_length\":61}},\"aws\":{\"table_name
\":\"scorekeep-user\",\"operation\":\"UpdateItem\",\"request_id\":
\"TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-user\"]},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",\"origin
\":\"AWS::DynamoDB::Table\"}"
                }
            ]
        }
    ],
    "UnprocessedTraceIds": []
}
```

Pour plus d'informations, consultez la section Utilisation de l'API AWS X-Ray avec la AWS CLI dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous BatchTracesGetà la section Référence des AWS CLI commandes.

`create-group`

L'exemple de code suivant montre comment utiliser`create-group`.

AWS CLI

Pour créer un groupe

L'`create-group`exemple suivant crée une ressource de groupe nomméeAdminGroup. Le groupe obtient une expression de filtre qui définit les critères du groupe sous la forme d'un segment lié à un service spécifique à l'origine d'une panne ou d'une erreur.

```
aws xray create-group \
   --group-name "AdminGroup" \
```

```
    --filter-expression "service(\"mydomain.com\") {fault OR error}"
```

Sortie :

```
{
    "GroupName": "AdminGroup",
    "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",
    "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous CreateGroupà la section Référence des AWS CLI commandes.

`create-sampling-rule`

L'exemple de code suivant montre comment utiliser`create-sampling-rule`.

AWS CLI

Pour créer une règle d'échantillonnage

L'`create-sampling-rule`exemple suivant crée une règle pour contrôler le comportement d'échantillonnage pour les applications instrumentées. Les règles sont fournies par un fichier JSON. La majorité des champs de règle d'échantillonnage sont obligatoires pour créer la règle.

```
aws xray create-sampling-rule \
    --cli-input-json file://9000-base-scorekeep.json
```

Contenu de `9000-base-scorekeep.json` :

```
{
    "SamplingRule": {
        "RuleName": "base-scorekeep",
        "ResourceARN": "*",
        "Priority": 9000,
        "FixedRate": 0.1,
        "ReservoirSize": 5,
        "ServiceName": "Scorekeep",
```

```
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1
    }
}
```

Sortie :

```
{
    "SamplingRuleRecord": {
        "SamplingRule": {
            "RuleName": "base-scorekeep",
            "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/base-
scorekeep",
            "ResourceARN": "*",
            "Priority": 9000,
            "FixedRate": 0.1,
            "ReservoirSize": 5,
            "ServiceName": "Scorekeep",
            "ServiceType": "*",
            "Host": "*",
            "HTTPMethod": "*",
            "URLPath": "*",
            "Version": 1,
            "Attributes": {}
        },
        "CreatedAt": 1530574410.0,
        "ModifiedAt": 1530574410.0
    }
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous CreateSamplingRuleà la section Référence des AWS CLI commandes.

`delete-group`

L'exemple de code suivant montre comment utiliser`delete-group`.

AWS CLI

Pour supprimer un groupe

L'`delete-group`exemple suivant supprime la ressource de groupe spécifiée.

```
aws xray delete-group \
    --group-name "AdminGroup" \
    --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous DeleteGroupà la section Référence des AWS CLI commandes.

`delete-sampling-rule`

L'exemple de code suivant montre comment utiliser`delete-sampling-rule`.

AWS CLI

Pour supprimer une règle d'échantillonnage

L'`delete-sampling-rule`exemple suivant supprime la règle d'échantillonnage spécifiée. Vous pouvez spécifier le groupe en utilisant son nom ou son ARN.

```
aws xray delete-sampling-rule \
    --rule-name polling-scorekeep
```

Sortie :

```
{
    "SamplingRuleRecord": {
        "SamplingRule": {
            "RuleName": "polling-scorekeep",
            "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/polling-
scorekeep",
            "ResourceARN": "*",
            "Priority": 5000,
```

```
            "FixedRate": 0.003,
            "ReservoirSize": 0,
            "ServiceName": "Scorekeep",
            "ServiceType": "*",
            "Host": "*",
            "HTTPMethod": "GET",
            "URLPath": "/api/state/*",
            "Version": 1,
            "Attributes": {}
        },
        "CreatedAt": 1530574399.0,
        "ModifiedAt": 1530574399.0
    }
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous DeleteSamplingRuleà la section Référence des AWS CLI commandes.

`get-encryption-config`

L'exemple de code suivant montre comment utiliser`get-encryption-config`.

AWS CLI

Pour récupérer la configuration de chiffrement

L'`get-encryption-config`exemple suivant permet de récupérer la configuration de chiffrement actuelle de vos données AWS X-Ray.

```
aws xray get-encryption-config
```

Sortie :

```
{
    "EncryptionConfig": {
        "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
        "Status": "ACTIVE",
        "Type": "NONE"
    }
```

```
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetEncryptionConfigà la section Référence des AWS CLI commandes.

`get-group`

L'exemple de code suivant montre comment utiliser`get-group`.

AWS CLI

Pour récupérer un groupe

L'`get-group`exemple suivant affiche les détails de la ressource de groupe spécifiée. Les détails incluent le nom du groupe, l'ARN du groupe et l'expression de filtre qui définit les critères pour ce groupe. Les groupes peuvent également être récupérés par ARN.

```
aws xray get-group \
    --group-name "AdminGroup"
```

Sortie :

```
{
    "Group": [
        {
            "GroupName": "AdminGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
            "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetGroupà la section Référence des AWS CLI commandes.

`get-groups`

L'exemple de code suivant montre comment utiliser`get-groups`.

AWS CLI

Pour récupérer tous les groupes

L'exemple suivant affiche les détails de tous les groupes actifs.

```
aws xray get-groups
```

Sortie :

```
{
    "Groups": [
        {
            "GroupName": "AdminGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
            "FilterExpression": "service(\"example.com\") {fault OR error}"
        },
        {
            "GroupName": "SDETGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
SDETGroup/987654321",
            "FilterExpression": "responsetime > 2"
        }
    ]
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetGroupsà la section Référence des AWS CLI commandes.

`get-sampling-rules`

L'exemple de code suivant montre comment utiliser`get-sampling-rules`.

AWS CLI

Pour récupérer toutes les règles d'échantillonnage

L'`get-sampling-rules`exemple suivant affiche les détails de toutes les règles d'échantillonnage disponibles. :

```
aws xray get-sampling-rules
```

Sortie :

```
{
    "SamplingRuleRecords": [
        {
            "SamplingRule": {
                "RuleName": "Default",
                "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/Default",
                "ResourceARN": "*",
                "Priority": 10000,
                "FixedRate": 0.01,
                "ReservoirSize": 0,
                "ServiceName": "*",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "*",
                "URLPath": "*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 0.0,
            "ModifiedAt": 1530558121.0
        },
        {
            "SamplingRule": {
                "RuleName": "base-scorekeep",
                "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/base-scorekeep",
                "ResourceARN": "*",
                "Priority": 9000,
                "FixedRate": 0.1,
                "ReservoirSize": 2,
                "ServiceName": "Scorekeep",
                "ServiceType": "*",
                "Host": "*",
```

```
                "HTTPMethod": "*",
                "URLPath": "*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 1530573954.0,
            "ModifiedAt": 1530920505.0
        },
        {
            "SamplingRule": {
                "RuleName": "polling-scorekeep",
                "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/polling-
scorekeep",
                "ResourceARN": "*",
                "Priority": 5000,
                "FixedRate": 0.003,
                "ReservoirSize": 0,
                "ServiceName": "Scorekeep",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "GET",
                "URLPath": "/api/state/*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 1530918163.0,
            "ModifiedAt": 1530918163.0
        }
    ]
}
```

Pour plus d'informations, consultez la section Utilisation des règles d'échantillonnage avec l'API X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetSamplingRulesà la section Référence des AWS CLI commandes.

`get-sampling-targets`

L'exemple de code suivant montre comment utiliser`get-sampling-targets`.

AWS CLI

Pour demander un quota d'échantillonnage

L'`get-sampling-targets`exemple suivant demande un quota d'échantillonnage pour les règles que le service utilise pour échantillonner les demandes. La réponse de AWS X-Ray inclut un quota qui peut être utilisé au lieu d'emprunter au réservoir.

```
aws xray get-sampling-targets \
    --sampling-statistics-documents '[ { "RuleName": "base-scorekeep", "ClientID":
 "ABCDEF1234567890ABCDEF10", "Timestamp": "2018-07-07T00:20:06, "RequestCount": 110,
 "SampledCount": 20, "BorrowCount": 10 }, { "RuleName": "polling-scorekeep", 31,
 "BorrowCount": 0 } ]'
```

Sortie :

```
{
    "SamplingTargetDocuments": [
        {
            "RuleName": "base-scorekeep",
            "FixedRate": 0.1,
            "ReservoirQuota": 2,
            "ReservoirQuotaTTL": 1530923107.0,
            "Interval": 10
        },
        {
            "RuleName": "polling-scorekeep",
            "FixedRate": 0.003,
            "ReservoirQuota": 0,
            "ReservoirQuotaTTL": 1530923107.0,
            "Interval": 10
        }
    ],
    "LastRuleModification": 1530920505.0,
    "UnprocessedStatistics": []
}
```

Pour plus d'informations, consultez la section Utilisation des règles d'échantillonnage avec l'API X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetSamplingTargetsà la section Référence des AWS CLI commandes.

`get-service-graph`

L'exemple de code suivant montre comment utiliser`get-service-graph`.

AWS CLI

Pour obtenir un graphique de service

L'exemple suivant affiche un document au cours d'une période spécifiée qui décrit les services traitant les demandes entrantes et les services en aval qu'ils appellent en conséquence. :

```
aws xray get-service-graph \
    --start-time 1568835392.0
    --end-time 1568835446.0
```

Sortie :

```
{
    "Services": [
        {
            "ReferenceId": 0,
            "Name": "Scorekeep",
            "Names": [
                "Scorekeep"
            ],
            "Root": true,
            "Type": "AWS::ElasticBeanstalk::Environment",
            "State": "active",
            "StartTime": 1568835392.0,
            "EndTime": 1568835446.0,
            "Edges": [
                {
                    "ReferenceId": 1,
                    "StartTime": 1568835392.0,
                    "EndTime": 1568835446.0,
                    "SummaryStatistics": {
                        "OkCount": 14,
                        "ErrorStatistics": {
                            "ThrottleCount": 0,
                            "OtherCount": 0,
                            "TotalCount": 0
                        },
                        "FaultStatistics": {
```

```
                        "OtherCount": 0,
                        "TotalCount": 0
                    },
                    "TotalCount": 14,
                    "TotalResponseTime": 0.13
                },
                "ResponseTimeHistogram": [
                    {
                        "Value": 0.008,
                        "Count": 1
                    },
                    {
                        "Value": 0.005,
                        "Count": 7
                    },
                    {
                        "Value": 0.009,
                        "Count": 1
                    },
                    {
                        "Value": 0.021,
                        "Count": 1
                    },
                    {
                        "Value": 0.038,
                        "Count": 1
                    },
                    {
                        "Value": 0.007,
                        "Count": 1
                    },
                    {
                        "Value": 0.006,
                        "Count": 2
                    }
                ],
                "Aliases": []
            },

            ... TRUNCATED FOR BREVITY ...

        ]
    }
],
```

```
    "StartTime": 1568835392.0,
    "EndTime": 1568835446.0,
    "ContainsOldGroupVersions": false
}
```

Pour plus d'informations, consultez la section Utilisation de l'API AWS X-Ray avec la AWS CLI dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetServiceGraphà la section Référence des AWS CLI commandes.

`get-trace-summaries`

L'exemple de code suivant montre comment utiliser`get-trace-summaries`.

AWS CLI

Pour obtenir un résumé des traces

L'`get-trace-summaries`exemple suivant récupère IDs les métadonnées des traces disponibles dans un laps de temps spécifié.

```
aws xray get-trace-summaries \
    --start-time 1568835392.0 \
    --end-time 1568835446.0
```

Sortie :

```
[
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/
VSAE93HF/GSSD2NTB/DP0PCC09",
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/
GCQ2B35P/FREELDFT/4LRE643M",
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/game/
VSAE93HF/GSSD2NTB/starttime/1568835513",
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/
move/4MQNA5NN/L99KK2RF/null"
]
```

Pour plus d'informations, consultez la section Utilisation de l'API AWS X-Ray avec la AWS CLI dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous GetTraceSummariesà la section Référence des AWS CLI commandes.

`put-encryption-config`

L'exemple de code suivant montre comment utiliser`put-encryption-config`.

AWS CLI

Pour mettre à jour la configuration de chiffrement

Ce qui suit`put-encryption-config``example updates the encryption configuration for AWS X-Ray data to use the default AWS managed KMS key ``aws/xray`.

```
aws xray put-encryption-config \
    --type KMS \
    --key-id alias/aws/xray
```

Sortie :

```
{
    "EncryptionConfig": {
        "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-
b0ea215cbba5",
        "Status": "UPDATING",
        "Type": "KMS"
    }
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous PutEncryptionConfigà la section Référence des AWS CLI commandes.

`put-trace-segments`

L'exemple de code suivant montre comment utiliser`put-trace-segments`.

AWS CLI

Pour télécharger un segment

L'`put-trace-segments`exemple suivant télécharge des documents segmentés dans AWS X-Ray. Le document de segment est consommé sous forme de liste de documents de segment JSON.

```
aws xray put-trace-segments \
    --trace-segment-documents "{\"id\":\"20312a0e2b8809f4\",\"name
\":\"DynamoDB\",\"trace_id\":\"1-5832862d-a43aafded3334a971fe312db\",
\"start_time\":1.479706157195E9,\"end_time\":1.479706157202E9,\"parent_id\":
\"79736b962fe3239e\",\"http\":{\"response\":{\"content_length\":60,\"status
\":200}},\"inferred\":true,\"aws\":{\"consistent_read\":false,\"table_name
\":\"scorekeep-session-xray\",\"operation\":\"GetItem\",\"request_id\":
\"SCAU23OM6M8FO38UASGC7785ARVV4KQNSO5AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-session-xray\"]},\"origin\":\"AWS::DynamoDB::Table\"}"
```

Sortie :

```
{
    "UnprocessedTraceSegments": []
}
```

Pour plus d'informations, consultez la section Sending Trace Data to AWS X-Ray dans le AWS X-Ray Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutTraceSegmentsà la section Référence des AWS CLI commandes.

`update-group`

L'exemple de code suivant montre comment utiliser`update-group`.

AWS CLI

Pour mettre à jour un groupe

L'`update-group`exemple suivant met à jour les critères d'acceptation des traces dans le groupe nomméAdminGroup. Vous pouvez spécifier le groupe souhaité en utilisant le nom ou l'ARN du groupe.

```
aws xray update-group \
    --group-name "AdminGroup" \
    --group-arn "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789" \
    --filter-expression "service(\"mydomain.com\") {fault}"
```

Sortie :

```
{
    "GroupName": "AdminGroup",
    "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
    "FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous UpdateGroupà la section Référence des AWS CLI commandes.

`update-sampling-rule`

L'exemple de code suivant montre comment utiliser`update-sampling-rule`.

AWS CLI

Pour mettre à jour une règle d'échantillonnage

L'`update-sampling-rule`exemple suivant modifie la configuration d'une règle d'échantillonnage. Les règles sont consommées à partir d'un fichier JSON. Seuls les champs mis à jour sont obligatoires.

```
aws xray update-sampling-rule \
   --cli-input-json file://1000-default.json
```

Contenu de `1000-default.json` :

```
{
    "SamplingRuleUpdate": {
        "RuleName": "Default",
        "FixedRate": 0.01,
```

```
        "ReservoirSize": 0
    }
}
```

Sortie :

```
{
    "SamplingRuleRecords": [
        {
            "SamplingRule": {
                "RuleName": "Default",
                "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/
Default",
                "ResourceARN": "*",
                "Priority": 10000,
                "FixedRate": 0.01,
                "ReservoirSize": 0,
                "ServiceName": "*",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "*",
                "URLPath": "*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 0.0,
            "ModifiedAt": 1529959993.0
        }
   ]
}
```

Pour plus d'informations, consultez la section Configuration des paramètres d'échantillonnage, de groupes et de chiffrement avec l'API AWS X-Ray dans le guide du développeur de AWS X-Ray.

- Pour plus de détails sur l'API, reportez-vous UpdateSamplingRuleà la section Référence des AWS CLI commandes.

AWS CLI avec des exemples de code de script Bash

Les exemples de code présentés dans cette rubrique vous montrent comment utiliser le script AWS Command Line Interface with Bash avec AWS.

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Certains services contiennent des exemples de catégories supplémentaires qui montrent comment tirer parti des bibliothèques ou des fonctions spécifiques au service.

Services

- AWS Batch exemples d'utilisation AWS CLI avec le script Bash
- AWS Cloud Map exemples d'utilisation AWS CLI avec le script Bash
- CloudFront exemples d'utilisation AWS CLI avec le script Bash
- Exemples AWS CLI d'utilisation de DynamoDB avec le script Bash
- EC2 Exemples d'utilisation d'Amazon AWS CLI avec le script Bash
- HealthImaging exemples d'utilisation AWS CLI avec le script Bash
- Exemples d'utilisation d'IAM AWS CLI avec le script Bash
- AWS KMS exemples d'utilisation AWS CLI avec le script Bash
- Exemples AWS CLI d'utilisation de Lightsail avec le script Bash
- Exemples d'utilisation d'Amazon S3 AWS CLI avec le script Bash
- AWS STS exemples d'utilisation AWS CLI avec le script Bash

AWS Batch exemples d'utilisation AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with AWS Batch.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Scénarios

Scénarios

Commencez avec Batch et Fargate

L'exemple de code suivant illustre comment :

- Création d'un rôle d'exécution IAM pour les tâches ECS
- Création d'un environnement informatique Fargate géré
- Création d'une file d'attente de tâches avec des paramètres de priorité
- Enregistrer une définition de tâche pour les charges de travail conteneurisées
- Soumettre et surveiller l'exécution d'une tâche par lots
- Afficher le résultat du travail dans CloudWatch les journaux
- Nettoyez les ressources dans le bon ordre de dépendance

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# AWS Batch Fargate Getting Started Script - Fixed Version
# This script demonstrates creating AWS Batch resources with Fargate orchestration
#
# HIGH SEVERITY FIXES APPLIED:
# 1. Added IAM role propagation delay after role creation
# 2. Added resource state validation before deletion attempts

set -e  # Exit on any error
```

```
# Configuration
SCRIPT_NAME="batch-fargate-tutorial"
LOG_FILE="${SCRIPT_NAME}-$(date +%Y%m%d-%H%M%S).log"
RANDOM_SUFFIX=$(openssl rand -hex 6)
COMPUTE_ENV_NAME="batch-fargate-compute-${RANDOM_SUFFIX}"
JOB_QUEUE_NAME="batch-fargate-queue-${RANDOM_SUFFIX}"
JOB_DEF_NAME="batch-fargate-jobdef-${RANDOM_SUFFIX}"
JOB_NAME="batch-hello-world-${RANDOM_SUFFIX}"
ROLE_NAME="BatchEcsTaskExecutionRole-${RANDOM_SUFFIX}"
TRUST_POLICY_FILE="batch-trust-policy-${RANDOM_SUFFIX}.json"

# Array to track created resources for cleanup
CREATED_RESOURCES=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    log "ERROR: Script failed at line $1"
    log "Attempting to clean up resources created so far..."
    cleanup_resources
    exit 1
}

# Set up error handling
trap 'handle_error $LINENO' ERR

# Function to wait for resource to be ready
wait_for_compute_env() {
    local env_name=$1
    log "Waiting for compute environment $env_name to be VALID..."

    while true; do
        local status=$(aws batch describe-compute-environments \
            --compute-environments "$env_name" \
            --query 'computeEnvironments[0].status' \
            --output text 2>/dev/null || echo "NOT_FOUND")

        if [ "$status" = "VALID" ]; then
            log "Compute environment $env_name is ready"
```

```
            break
        elif [ "$status" = "INVALID" ] || [ "$status" = "NOT_FOUND" ]; then
            log "ERROR: Compute environment $env_name failed to create properly"
            return 1
        fi

        log "Compute environment status: $status. Waiting 10 seconds..."
        sleep 10
    done
}

# Function to wait for job queue to be ready
wait_for_job_queue() {
    local queue_name=$1
    log "Waiting for job queue $queue_name to be VALID..."

    while true; do
        local state=$(aws batch describe-job-queues \
            --job-queues "$queue_name" \
            --query 'jobQueues[0].state' \
            --output text 2>/dev/null || echo "NOT_FOUND")

        if [ "$state" = "ENABLED" ]; then
            log "Job queue $queue_name is ready"
            break
        elif [ "$state" = "DISABLED" ] || [ "$state" = "NOT_FOUND" ]; then
            log "ERROR: Job queue $queue_name failed to create properly"
            return 1
        fi

        log "Job queue state: $state. Waiting 10 seconds..."
        sleep 10
    done
}

# Function to wait for job completion
wait_for_job() {
    local job_id=$1
    log "Waiting for job $job_id to complete..."

    while true; do
        local status=$(aws batch describe-jobs \
            --jobs "$job_id" \
            --query 'jobs[0].status' \
```

```
            --output text 2>/dev/null || echo "NOT_FOUND")

        if [ "$status" = "SUCCEEDED" ]; then
            log "Job $job_id completed successfully"
            break
        elif [ "$status" = "FAILED" ]; then
            log "ERROR: Job $job_id failed"
            return 1
        fi

        log "Job status: $status. Waiting 30 seconds..."
        sleep 30
    done
}

# FIXED: Added function to wait for resource state before deletion
wait_for_resource_state() {
    local resource_type=$1
    local resource_name=$2
    local expected_state=$3
    local max_attempts=30
    local attempt=0

    log "Waiting for $resource_type $resource_name to reach state: $expected_state"

    while [ $attempt -lt $max_attempts ]; do
        local current_state=""

        case $resource_type in
            "JOB_QUEUE")
                current_state=$(aws batch describe-job-queues \
                    --job-queues "$resource_name" \
                    --query 'jobQueues[0].state' \
                    --output text 2>/dev/null || echo "NOT_FOUND")
                ;;
            "COMPUTE_ENV")
                current_state=$(aws batch describe-compute-environments \
                    --compute-environments "$resource_name" \
                    --query 'computeEnvironments[0].status' \
                    --output text 2>/dev/null || echo "NOT_FOUND")
                ;;
        esac

        if [ "$current_state" = "$expected_state" ]; then
```

```
            log "$resource_type $resource_name is now in state: $expected_state"
            return 0
        fi

        log "$resource_type $resource_name state: $current_state (waiting for
 $expected_state)"
        sleep 10
        ((attempt++))
    done

    log "WARNING: $resource_type $resource_name did not reach expected state after
 $max_attempts attempts"
    return 1
}

# Cleanup function
cleanup_resources() {
    log "Starting cleanup of created resources..."

    # Clean up in reverse order of creation
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        local resource="${CREATED_RESOURCES[i]}"
        local resource_type=$(echo "$resource" | cut -d: -f1)
        local resource_name=$(echo "$resource" | cut -d: -f2)

        log "Cleaning up $resource_type: $resource_name"

        case $resource_type in
            "JOB_QUEUE")
                # FIXED: Validate state before deletion
                aws batch update-job-queue --job-queue "$resource_name" --state
 DISABLED 2>/dev/null || true
                wait_for_resource_state "JOB_QUEUE" "$resource_name" "DISABLED" ||
 true
                aws batch delete-job-queue --job-queue "$resource_name" 2>/dev/null
 || true
                ;;
            "COMPUTE_ENV")
                # FIXED: Validate state before deletion
                aws batch update-compute-environment --compute-environment
 "$resource_name" --state DISABLED 2>/dev/null || true
                wait_for_resource_state "COMPUTE_ENV" "$resource_name" "DISABLED" ||
 true
```

```
                aws batch delete-compute-environment --compute-environment
 "$resource_name" 2>/dev/null || true
                ;;
            "IAM_ROLE")
                aws iam detach-role-policy --role-name "$resource_name" --policy-arn
 "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/dev/null
 || true
                aws iam delete-role --role-name "$resource_name" 2>/dev/null || true
                ;;
            "FILE")
                rm -f "$resource_name" 2>/dev/null || true
                ;;
        esac
    done

    log "Cleanup completed"
}

# Main script execution
main() {
    log "Starting AWS Batch Fargate tutorial script - Fixed Version"
    log "Log file: $LOG_FILE"

    # Get AWS account ID
    log "Getting AWS account ID..."
    ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
    log "Account ID: $ACCOUNT_ID"

    # Get default VPC and subnets
    log "Getting default VPC and subnets..."
    DEFAULT_VPC=$(aws ec2 describe-vpcs \
        --filters "Name=is-default,Values=true" \
        --query 'Vpcs[0].VpcId' \
        --output text)

    if [ "$DEFAULT_VPC" = "None" ] || [ "$DEFAULT_VPC" = "null" ]; then
        log "ERROR: No default VPC found. Please create a VPC first."
        exit 1
    fi

    log "Default VPC: $DEFAULT_VPC"

    # Get subnets in the default VPC
    SUBNETS=$(aws ec2 describe-subnets \
```

```
        --filters "Name=vpc-id,Values=$DEFAULT_VPC" \
        --query 'Subnets[*].SubnetId' \
        --output text)

    if [ -z "$SUBNETS" ]; then
        log "ERROR: No subnets found in default VPC"
        exit 1
    fi

    # Convert tab/space-separated subnets to JSON array format
    SUBNET_ARRAY=$(echo "$SUBNETS" | tr '\t ' '\n' | sed 's/^/"/;s/$/"/' | paste -sd
 ',' -)
    log "Subnets: $SUBNETS"
    log "Subnet array: [$SUBNET_ARRAY]"

    # Get default security group for the VPC
    DEFAULT_SG=$(aws ec2 describe-security-groups \
        --filters "Name=vpc-id,Values=$DEFAULT_VPC" "Name=group-name,Values=default"
 \
        --query 'SecurityGroups[0].GroupId' \
        --output text)

    if [ "$DEFAULT_SG" = "None" ] || [ "$DEFAULT_SG" = "null" ]; then
        log "ERROR: No default security group found in VPC"
        exit 1
    fi

    log "Default security group: $DEFAULT_SG"

    # Step 1: Create IAM execution role
    log "Step 1: Creating IAM execution role..."

    # Create trust policy document
    cat > "$TRUST_POLICY_FILE" << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
EOF
    CREATED_RESOURCES+=("FILE:$TRUST_POLICY_FILE")

    # Create the role
    aws iam create-role \
        --role-name "$ROLE_NAME" \
        --assume-role-policy-document "file://$TRUST_POLICY_FILE"
    CREATED_RESOURCES+=("IAM_ROLE:$ROLE_NAME")

    # Attach policy
    aws iam attach-role-policy \
        --role-name "$ROLE_NAME" \
        --policy-arn "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"

    log "IAM role created: $ROLE_NAME"

    # FIXED: Wait for IAM role propagation
    log "Waiting for IAM role propagation (15 seconds)..."
    sleep 15

    # Step 2: Create compute environment
    log "Step 2: Creating Fargate compute environment..."

    aws batch create-compute-environment \
        --compute-environment-name "$COMPUTE_ENV_NAME" \
        --type MANAGED \
        --state ENABLED \
        --compute-resources "{
            \"type\": \"FARGATE\",
            \"maxvCpus\": 256,
            \"subnets\": [$SUBNET_ARRAY],
            \"securityGroupIds\": [\"$DEFAULT_SG\"]
        }"
    CREATED_RESOURCES+=("COMPUTE_ENV:$COMPUTE_ENV_NAME")

    # Wait for compute environment to be ready
    wait_for_compute_env "$COMPUTE_ENV_NAME"

    # Step 3: Create job queue
    log "Step 3: Creating job queue..."
```

```
    aws batch create-job-queue \
        --job-queue-name "$JOB_QUEUE_NAME" \
        --state ENABLED \
        --priority 900 \
        --compute-environment-order order=1,computeEnvironment="$COMPUTE_ENV_NAME"
    CREATED_RESOURCES+=("JOB_QUEUE:$JOB_QUEUE_NAME")

    # Wait for job queue to be ready
    wait_for_job_queue "$JOB_QUEUE_NAME"

    # Step 4: Create job definition
    log "Step 4: Creating job definition..."

    aws batch register-job-definition \
        --job-definition-name "$JOB_DEF_NAME" \
        --type container \
        --platform-capabilities FARGATE \
        --container-properties "{
            \"image\": \"busybox\",
            \"resourceRequirements\": [
                {\"type\": \"VCPU\", \"value\": \"0.25\"},
                {\"type\": \"MEMORY\", \"value\": \"512\"}
            ],
            \"command\": [\"echo\", \"hello world\"],
            \"networkConfiguration\": {
                \"assignPublicIp\": \"ENABLED\"
            },
            \"executionRoleArn\": \"arn:aws:iam::${ACCOUNT_ID}:role/${ROLE_NAME}\"
        }"

    log "Job definition created: $JOB_DEF_NAME"

    # Step 5: Submit job
    log "Step 5: Submitting job..."

    JOB_ID=$(aws batch submit-job \
        --job-name "$JOB_NAME" \
        --job-queue "$JOB_QUEUE_NAME" \
        --job-definition "$JOB_DEF_NAME" \
        --query 'jobId' \
        --output text)

    log "Job submitted with ID: $JOB_ID"
```

```
# Step 6: Wait for job completion and view output
log "Step 6: Waiting for job completion..."
wait_for_job "$JOB_ID"

# Get log stream name
log "Getting job logs..."
LOG_STREAM=$(aws batch describe-jobs \
    --jobs "$JOB_ID" \
    --query 'jobs[0].attempts[0].taskProperties.containers[0].logStreamName' \
    --output text)

if [ "$LOG_STREAM" != "None" ] && [ "$LOG_STREAM" != "null" ]; then
    log "Log stream: $LOG_STREAM"
    log "Job output:"
    aws logs get-log-events \
        --log-group-name "/aws/batch/job" \
        --log-stream-name "$LOG_STREAM" \
        --query 'events[*].message' \
        --output text | tee -a "$LOG_FILE"
else
    log "No log stream available for job"
fi

log "Tutorial completed successfully!"

# Show created resources
echo ""
echo "==========================================="
echo "CREATED RESOURCES"
echo "==========================================="
echo "The following resources were created:"
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "  - $resource"
done
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    cleanup_resources
    log "All resources have been cleaned up"
```

```
    else
        log "Resources left intact. You can clean them up manually later."
        echo "To clean up manually, run the following commands:"
        echo "aws batch update-job-queue --job-queue $JOB_QUEUE_NAME --state
 DISABLED"
        echo "aws batch delete-job-queue --job-queue $JOB_QUEUE_NAME"
        echo "aws batch update-compute-environment --compute-environment
 $COMPUTE_ENV_NAME --state DISABLED"
        echo "aws batch delete-compute-environment --compute-environment
 $COMPUTE_ENV_NAME"
        echo "aws iam detach-role-policy --role-name $ROLE_NAME --policy-arn
 arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
        echo "aws iam delete-role --role-name $ROLE_NAME"
    fi
}

# Run main function
main "$@"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateComputeEnvironment
  - CreateJobQueue
  - DeleteComputeEnvironment
  - DeleteJobQueue
  - DescribeComputeEnvironments
  - DescribeJobQueues
  - DescribeJobs
  - RegisterJobDefinition
  - SubmitJob
  - UpdateComputeEnvironment
  - UpdateJobQueue

AWS Cloud Map exemples d'utilisation AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with AWS Cloud Map.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Scénarios

Scénarios

Attributs personnalisés de Cloud Map

L'exemple de code suivant illustre comment :

- Création d'un espace de noms HTTP pour la découverte de services basée sur une API
- Créez une table DynamoDB et enregistrez-la en tant que service de données avec des attributs personnalisés
- Création de fonctions Lambda pour lire et écrire des données
- Enregistrez les fonctions Lambda en tant qu'instances de service avec des attributs personnalisés pour une découverte basée sur l'action
- Créez des applications clientes qui découvrent les services à l'aide d'attributs personnalisés
- Nettoyez toutes les ressources, y compris les fonctions Lambda, la table DynamoDB et les services Cloud Map

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# AWS Cloud Map Tutorial Script
# This script demonstrates how to use AWS Cloud Map for service discovery with
 custom attributes

# Set up logging
LOG_FILE="cloudmap-tutorial.log"
echo "AWS Cloud Map Tutorial Script" > $LOG_FILE
echo "Started at $(date)" >> $LOG_FILE

# Array to track created resources for cleanup
CREATED_RESOURCES=()

# Function to log commands and their output
log_cmd() {
  echo "$ $1" | tee -a $LOG_FILE
  eval "$1" | tee -a $LOG_FILE
}

# Function to handle errors
handle_error() {
  local LINE=$1
  echo "An error occurred at line $LINE" | tee -a $LOG_FILE
  echo "Resources created so far:" | tee -a $LOG_FILE
  for resource in "${CREATED_RESOURCES[@]}"; do
    echo "- $resource" | tee -a $LOG_FILE
  done
  echo "Attempting to clean up resources..." | tee -a $LOG_FILE
  cleanup
  exit 1
}

# Set up error handling
trap 'handle_error $LINENO' ERR

# Helper function to wait for Cloud Map operations to complete
wait_for_operation() {
  local OPERATION_ID=$1
  local TIMEOUT=300  # 5 minutes timeout
  local START_TIME=$(date +%s)

  while true; do
```

```
    local STATUS=$(aws servicediscovery get-operation --operation-id $OPERATION_ID
 --query 'Operation.Status' --output text)

    if [ "$STATUS" == "SUCCESS" ]; then
      echo "Operation completed successfully" | tee -a $LOG_FILE
      break
    elif [ "$STATUS" == "FAIL" ]; then
      echo "Operation failed" | tee -a $LOG_FILE
      return 1
    fi

    local CURRENT_TIME=$(date +%s)
    if [ $((CURRENT_TIME - START_TIME)) -gt $TIMEOUT ]; then
      echo "Operation timed out" | tee -a $LOG_FILE
      return 1
    fi

    sleep 5
  done

  return 0
}

# Function to clean up resources
cleanup() {
  echo "Cleaning up resources..." | tee -a $LOG_FILE

  # Reverse the order of created resources for proper deletion
  for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
    resource="${CREATED_RESOURCES[$i]}"
    echo "Deleting $resource..." | tee -a $LOG_FILE

    if [[ $resource == "instance:"* ]]; then
      # Extract service ID and instance ID
      SERVICE_ID=$(echo $resource | cut -d':' -f2)
      INSTANCE_ID=$(echo $resource | cut -d':' -f3)

      # Check if instance exists before trying to deregister
      INSTANCE_EXISTS=$(aws servicediscovery list-instances --service-id $SERVICE_ID
 --query "Instances[?Id=='$INSTANCE_ID'].Id" --output text 2>/dev/null || echo "")
      if [[ -n "$INSTANCE_EXISTS" ]]; then
        OPERATION_ID=$(aws servicediscovery deregister-instance --service-id
 $SERVICE_ID --instance-id $INSTANCE_ID --query 'OperationId' --output text)
```

```
        # Wait for deregistration to complete
        echo "Waiting for instance deregistration to complete..." | tee -a $LOG_FILE
        wait_for_operation $OPERATION_ID
      else
        echo "Instance $INSTANCE_ID already deregistered" | tee -a $LOG_FILE
      fi
    elif [[ $resource == "lambda:"* ]]; then
      # Extract function name
      FUNCTION_NAME=$(echo $resource | cut -d':' -f2)
      aws lambda delete-function --function-name $FUNCTION_NAME
    elif [[ $resource == "role:"* ]]; then
      # Extract role name
      ROLE_NAME=$(echo $resource | cut -d':' -f2)

      # Detach all policies first
      for POLICY_ARN in $(aws iam list-attached-role-policies --role-name $ROLE_NAME
 --query 'AttachedPolicies[*].PolicyArn' --output text); do
        aws iam detach-role-policy --role-name $ROLE_NAME --policy-arn $POLICY_ARN
      done

      # Delete the role
      aws iam delete-role --role-name $ROLE_NAME
    elif [[ $resource == "dynamodb:"* ]]; then
      # Extract table name
      TABLE_NAME=$(echo $resource | cut -d':' -f2)
      aws dynamodb delete-table --table-name $TABLE_NAME

      # Wait for table deletion to complete
      echo "Waiting for DynamoDB table deletion to complete..." | tee -a $LOG_FILE
      aws dynamodb wait table-not-exists --table-name $TABLE_NAME
    fi
  done

  # Handle services separately to ensure all instances are deregistered first
  for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
    resource="${CREATED_RESOURCES[$i]}"
    if [[ $resource == "service:"* ]]; then
      # Extract service ID
      SERVICE_ID=$(echo $resource | cut -d':' -f2)
      echo "Deleting service $SERVICE_ID..." | tee -a $LOG_FILE

      # Make sure all instances are deregistered
      INSTANCES=$(aws servicediscovery list-instances --service-id $SERVICE_ID --
query 'Instances[*].Id' --output text)
```

```
      if [[ -n "$INSTANCES" ]]; then
        echo "Service still has instances. Waiting before deletion..." | tee -a
 $LOG_FILE
        sleep 10
      fi

      # Try to delete the service
      aws servicediscovery delete-service --id $SERVICE_ID
      sleep 5
    fi
  done

  # Handle namespaces last to ensure all services are deleted first
  for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
    resource="${CREATED_RESOURCES[$i]}"
    if [[ $resource == "namespace:"* ]]; then
      # Extract namespace ID
      NAMESPACE_ID=$(echo $resource | cut -d':' -f2)
      echo "Deleting namespace $NAMESPACE_ID..." | tee -a $LOG_FILE

      # Check if namespace still has services
      SERVICES=$(aws servicediscovery list-services --filters
 "Name=NAMESPACE_ID,Values=$NAMESPACE_ID,Condition=EQ" --query 'Services[*].Id' --
output text)
      if [[ -n "$SERVICES" ]]; then
        echo "Namespace still has services. Deleting them first..." | tee -a
 $LOG_FILE
        for SERVICE_ID in $SERVICES; do
          echo "Deleting service $SERVICE_ID..." | tee -a $LOG_FILE
          aws servicediscovery delete-service --id $SERVICE_ID
        done
        sleep 5
      fi

      # Try to delete the namespace
      OPERATION_ID=$(aws servicediscovery delete-namespace --id $NAMESPACE_ID --
query 'OperationId' --output text 2>/dev/null || echo "")
      if [[ -n "$OPERATION_ID" ]]; then
        echo "Waiting for namespace deletion to complete..." | tee -a $LOG_FILE
        wait_for_operation $OPERATION_ID
      else
        echo "Failed to delete namespace or namespace already deleted" | tee -a
 $LOG_FILE
      fi
```

```
    fi
  done

  echo "Cleanup complete" | tee -a $LOG_FILE
}

# Step 1: Create an AWS Cloud Map namespace
echo "Step 1: Creating AWS Cloud Map namespace..." | tee -a $LOG_FILE

# Check if namespace already exists
NAMESPACE_ID=$(aws servicediscovery list-namespaces --query "Namespaces[?
Name=='cloudmap-tutorial'].Id" --output text)

if [[ -z "$NAMESPACE_ID" || "$NAMESPACE_ID" == "None" ]]; then
  log_cmd "aws servicediscovery create-http-namespace --name cloudmap-tutorial --
creator-request-id namespace-request"
  OPERATION_ID=$(aws servicediscovery create-http-namespace --name cloudmap-tutorial
 --creator-request-id namespace-request --query 'OperationId' --output text)

  # Wait for namespace creation to complete
  echo "Waiting for namespace creation to complete..." | tee -a $LOG_FILE
  wait_for_operation $OPERATION_ID

  # Get the namespace ID
  NAMESPACE_ID=$(aws servicediscovery list-namespaces --query "Namespaces[?
Name=='cloudmap-tutorial'].Id" --output text)
  echo "Namespace created with ID: $NAMESPACE_ID" | tee -a $LOG_FILE
else
  echo "Namespace cloudmap-tutorial already exists with ID: $NAMESPACE_ID" | tee -a
 $LOG_FILE
fi

CREATED_RESOURCES+=("namespace:$NAMESPACE_ID")

# Step 2: Create a DynamoDB table
echo "Step 2: Creating DynamoDB table..." | tee -a $LOG_FILE

# Check if table already exists
TABLE_EXISTS=$(aws dynamodb describe-table --table-name cloudmap 2>&1 || echo
 "NOT_EXISTS")

if [[ $TABLE_EXISTS == *"ResourceNotFoundException"* || $TABLE_EXISTS ==
 "NOT_EXISTS" ]]; then
```

```
  log_cmd "aws dynamodb create-table --table-name cloudmap --attribute-definitions
 AttributeName=id,AttributeType=S --key-schema AttributeName=id,KeyType=HASH --
billing-mode PAY_PER_REQUEST"

  # Wait for DynamoDB table to become active
  echo "Waiting for DynamoDB table to become active..." | tee -a $LOG_FILE
  aws dynamodb wait table-exists --table-name cloudmap
else
  echo "DynamoDB table cloudmap already exists" | tee -a $LOG_FILE
fi

CREATED_RESOURCES+=("dynamodb:cloudmap")

# Step 3: Create an AWS Cloud Map data service
echo "Step 3: Creating AWS Cloud Map data service..." | tee -a $LOG_FILE

# Get all services in the namespace
echo "Listing all services in namespace $NAMESPACE_ID..." | tee -a $LOG_FILE
SERVICES=$(aws servicediscovery list-services --filters "Name=NAMESPACE_ID,Values=
$NAMESPACE_ID,Condition=EQ" --query 'Services[*].[Id,Name]' --output text)
echo "Services found: $SERVICES" | tee -a $LOG_FILE

# Check if data service already exists
DATA_SERVICE_ID=""
while read -r id name || [[ -n "$id" ]]; do
  echo "Checking service: ID=$id, Name=$name" | tee -a $LOG_FILE
  if [[ "$name" == "data-service" ]]; then
    DATA_SERVICE_ID="$id"
    break
  fi
done <<< "$SERVICES"

if [[ -z "$DATA_SERVICE_ID" ]]; then
  echo "Data service does not exist, creating it..." | tee -a $LOG_FILE
  # Create the service and capture the ID directly
  echo "$ aws servicediscovery create-service --name data-service --namespace-id
 $NAMESPACE_ID --creator-request-id data-service-request" | tee -a $LOG_FILE
  CREATE_OUTPUT=$(aws servicediscovery create-service --name data-service --
namespace-id $NAMESPACE_ID --creator-request-id data-service-request)
  echo "$CREATE_OUTPUT" | tee -a $LOG_FILE

  # Extract the service ID using AWS CLI query
```

```
  DATA_SERVICE_ID=$(aws servicediscovery list-services --filters
 "Name=NAMESPACE_ID,Values=$NAMESPACE_ID,Condition=EQ" --query "Services[?
Name=='data-service'].Id" --output text)
  echo "Data service created with ID: $DATA_SERVICE_ID" | tee -a $LOG_FILE
else
  echo "Data service already exists with ID: $DATA_SERVICE_ID" | tee -a $LOG_FILE
fi

CREATED_RESOURCES+=("service:$DATA_SERVICE_ID")

# Register DynamoDB table as a service instance
echo "Registering DynamoDB table as a service instance..." | tee -a $LOG_FILE

# Check if instance already exists
INSTANCE_EXISTS=$(aws servicediscovery list-instances --service-id $DATA_SERVICE_ID
 --query "Instances[?Id=='data-instance'].Id" --output text)

if [[ -z "$INSTANCE_EXISTS" ]]; then
  log_cmd "aws servicediscovery register-instance --service-id $DATA_SERVICE_ID --
instance-id data-instance --attributes tablename=cloudmap,region=$(aws configure get
 region)"
  OPERATION_ID=$(aws servicediscovery register-instance --service-
id $DATA_SERVICE_ID --instance-id data-instance --attributes
 tablename=cloudmap,region=$(aws configure get region) --query 'OperationId' --
output text)

  # Wait for instance registration to complete
  echo "Waiting for instance registration to complete..." | tee -a $LOG_FILE
  wait_for_operation $OPERATION_ID
else
  echo "Instance data-instance already exists" | tee -a $LOG_FILE
fi

CREATED_RESOURCES+=("instance:$DATA_SERVICE_ID:data-instance")

# Step 4: Create an IAM role for Lambda
echo "Step 4: Creating IAM role for Lambda..." | tee -a $LOG_FILE

# Create a trust policy for Lambda
cat > lambda-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Check if role already exists
echo "Checking if IAM role already exists..." | tee -a $LOG_FILE
ROLE_EXISTS=$(aws iam get-role --role-name cloudmap-tutorial-role 2>&1 || echo
 "NOT_EXISTS")

if [[ $ROLE_EXISTS == *"NoSuchEntity"* || $ROLE_EXISTS == "NOT_EXISTS" ]]; then
    log_cmd "aws iam create-role --role-name cloudmap-tutorial-role --assume-role-
policy-document file://lambda-trust-policy.json"
else
    echo "Role cloudmap-tutorial-role already exists, using existing role" | tee -a
 $LOG_FILE
fi

# FIXED: Create a custom policy with least privilege instead of using
 PowerUserAccess
cat > cloudmap-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:GetItem",
        "dynamodb:Scan"
```

```
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/cloudmap"
    },
    {
      "Effect": "Allow",
      "Action": [
        "servicediscovery:DiscoverInstances"
      ],
      "Resource": "*"
    }
  ]
}
EOF

# Check if policy already exists
POLICY_ARN=$(aws iam list-policies --query "Policies[?
PolicyName=='CloudMapTutorialPolicy'].Arn" --output text)

if [[ -z "$POLICY_ARN" ]]; then
  echo "Creating CloudMapTutorialPolicy..." | tee -a $LOG_FILE
  echo "$ aws iam create-policy --policy-name CloudMapTutorialPolicy --policy-
document file://cloudmap-policy.json" | tee -a $LOG_FILE
  CREATE_OUTPUT=$(aws iam create-policy --policy-name CloudMapTutorialPolicy --
policy-document file://cloudmap-policy.json)
  echo "$CREATE_OUTPUT" | tee -a $LOG_FILE
  POLICY_ARN=$(aws iam list-policies --query "Policies[?
PolicyName=='CloudMapTutorialPolicy'].Arn" --output text)
else
  echo "Policy CloudMapTutorialPolicy already exists with ARN: $POLICY_ARN" | tee -a
 $LOG_FILE
fi

echo "$ aws iam attach-role-policy --role-name cloudmap-tutorial-role --policy-arn
 $POLICY_ARN" | tee -a $LOG_FILE
aws iam attach-role-policy --role-name cloudmap-tutorial-role --policy-arn
 $POLICY_ARN | tee -a $LOG_FILE

echo "$ aws iam attach-role-policy --role-name cloudmap-tutorial-role --policy-
arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole" | tee -a
 $LOG_FILE
aws iam attach-role-policy --role-name cloudmap-tutorial-role --policy-arn
 arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole | tee -a $LOG_FILE

# Wait for role to propagate
```

```
echo "Waiting for IAM role to propagate..." | tee -a $LOG_FILE
sleep 10

ROLE_ARN=$(aws iam get-role --role-name cloudmap-tutorial-role --query 'Role.Arn' --
output text)
CREATED_RESOURCES+=("role:cloudmap-tutorial-role")

# Step 5: Create an AWS Cloud Map app service
echo "Step 5: Creating AWS Cloud Map app service..." | tee -a $LOG_FILE

# Get all services in the namespace
SERVICES=$(aws servicediscovery list-services --filters "Name=NAMESPACE_ID,Values=
$NAMESPACE_ID,Condition=EQ" --query 'Services[*].[Id,Name]' --output text)

# Check if app service already exists
APP_SERVICE_ID=""
while read -r id name || [[ -n "$id" ]]; do
  if [[ "$name" == "app-service" ]]; then
    APP_SERVICE_ID="$id"
    break
  fi
done <<< "$SERVICES"

if [[ -z "$APP_SERVICE_ID" ]]; then
  echo "App service does not exist, creating it..." | tee -a $LOG_FILE
  # Create the service and capture the ID directly
  echo "$ aws servicediscovery create-service --name app-service --namespace-id
 $NAMESPACE_ID --creator-request-id app-service-request" | tee -a $LOG_FILE
  CREATE_OUTPUT=$(aws servicediscovery create-service --name app-service --
namespace-id $NAMESPACE_ID --creator-request-id app-service-request)
  echo "$CREATE_OUTPUT" | tee -a $LOG_FILE

  # Extract the service ID using AWS CLI query
  APP_SERVICE_ID=$(aws servicediscovery list-services --filters
 "Name=NAMESPACE_ID,Values=$NAMESPACE_ID,Condition=EQ" --query "Services[?
Name=='app-service'].Id" --output text)
  echo "App service created with ID: $APP_SERVICE_ID" | tee -a $LOG_FILE
else
  echo "App service already exists with ID: $APP_SERVICE_ID" | tee -a $LOG_FILE
fi

CREATED_RESOURCES+=("service:$APP_SERVICE_ID")

# Step 6: Create a Lambda function to write data
```

```
echo "Step 6: Creating Lambda function to write data..." | tee -a $LOG_FILE

# Create Lambda function code
cat > writefunction.py << EOF
import boto3
import json
import random

def lambda_handler(event, context):
    # Use AWS Cloud Map to discover the DynamoDB table
    serviceclient = boto3.client('servicediscovery')

    # Discover the data service instance
    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='data-service'
    )

    # Extract table name and region from the instance attributes
    tablename = response['Instances'][0]['Attributes']['tablename']
    region = response['Instances'][0]['Attributes']['region']

    # Create DynamoDB client in the specified region
    dynamodb = boto3.resource('dynamodb', region_name=region)
    table = dynamodb.Table(tablename)

    # Write data to the table
    table.put_item(
        Item={
            'id': str(random.randint(1,100)),
            'todo': event
        }
    )

    return {
        'statusCode': 200,
        'body': json.dumps('Data written successfully!')
    }
EOF

# Zip the function code
log_cmd "zip writefunction.zip writefunction.py"

# Create the Lambda function
```

```
FUNCTION_EXISTS=$(aws lambda list-functions --query "Functions[?
FunctionName=='writefunction'].FunctionName" --output text)
if [[ -z "$FUNCTION_EXISTS" ]]; then
  log_cmd "aws lambda create-function --function-name writefunction --runtime
 python3.12 --role $ROLE_ARN --handler writefunction.lambda_handler --zip-file
 fileb://writefunction.zip --architectures x86_64"

  # Wait for the Lambda function to be active before updating
  echo "Waiting for Lambda function to become active..." | tee -a $LOG_FILE
  function_state="Pending"
  while [ "$function_state" == "Pending" ]; do
      sleep 5
      function_state=$(aws lambda get-function --function-name writefunction --query
 'Configuration.State' --output text)
      echo "Current function state: $function_state" | tee -a $LOG_FILE
  done

  # Update the function timeout
  log_cmd "aws lambda update-function-configuration --function-name writefunction --
timeout 5"
else
  echo "Lambda function writefunction already exists" | tee -a $LOG_FILE
fi
CREATED_RESOURCES+=("lambda:writefunction")

# Step 7: Register the Lambda write function as an AWS Cloud Map service instance
echo "Step 7: Registering Lambda write function as a service instance..." | tee -a
 $LOG_FILE

# Check if instance already exists
INSTANCE_EXISTS=$(aws servicediscovery list-instances --service-id $APP_SERVICE_ID
 --query "Instances[?Id=='write-instance'].Id" --output text)

if [[ -z "$INSTANCE_EXISTS" ]]; then
  log_cmd "aws servicediscovery register-instance --service-id $APP_SERVICE_ID --
instance-id write-instance --attributes action=write,functionname=writefunction"
  OPERATION_ID=$(aws servicediscovery register-instance --service-id $APP_SERVICE_ID
 --instance-id write-instance --attributes action=write,functionname=writefunction
 --query 'OperationId' --output text)

  # Wait for instance registration to complete
  echo "Waiting for write instance registration to complete..." | tee -a $LOG_FILE
  wait_for_operation $OPERATION_ID
else
```

```
  echo "Instance write-instance already exists" | tee -a $LOG_FILE
fi

CREATED_RESOURCES+=("instance:$APP_SERVICE_ID:write-instance")

# Step 8: Create a Lambda function to read data
echo "Step 8: Creating Lambda function to read data..." | tee -a $LOG_FILE

# Create Lambda function code
cat > readfunction.py << EOF
import boto3
import json

def lambda_handler(event, context):
    # Use AWS Cloud Map to discover the DynamoDB table
    serviceclient = boto3.client('servicediscovery')

    # Discover the data service instance
    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='data-service'
    )

    # Extract table name and region from the instance attributes
    tablename = response['Instances'][0]['Attributes']['tablename']
    region = response['Instances'][0]['Attributes']['region']

    # Create DynamoDB client in the specified region
    dynamodb = boto3.resource('dynamodb', region_name=region)
    table = dynamodb.Table(tablename)

    # Read data from the table
    response = table.scan()

    return {
        'statusCode': 200,
        'body': json.dumps(response['Items'])
    }
EOF

# Zip the function code
log_cmd "zip readfunction.zip readfunction.py"

# Create the Lambda function
```

```
FUNCTION_EXISTS=$(aws lambda list-functions --query "Functions[?
FunctionName=='readfunction'].FunctionName" --output text)
if [[ -z "$FUNCTION_EXISTS" ]]; then
  log_cmd "aws lambda create-function --function-name readfunction --runtime
 python3.12 --role $ROLE_ARN --handler readfunction.lambda_handler --zip-file
 fileb://readfunction.zip --architectures x86_64"

  # Wait for the Lambda function to be active before updating
  echo "Waiting for Lambda function to become active..." | tee -a $LOG_FILE
  function_state="Pending"
  while [ "$function_state" == "Pending" ]; do
      sleep 5
      function_state=$(aws lambda get-function --function-name readfunction --query
 'Configuration.State' --output text)
      echo "Current function state: $function_state" | tee -a $LOG_FILE
  done

  # Update the function timeout
  log_cmd "aws lambda update-function-configuration --function-name readfunction --
timeout 5"
else
  echo "Lambda function readfunction already exists" | tee -a $LOG_FILE
fi
CREATED_RESOURCES+=("lambda:readfunction")

# Step 9: Register the Lambda read function as an AWS Cloud Map service instance
echo "Step 9: Registering Lambda read function as a service instance..." | tee -a
 $LOG_FILE

# Check if instance already exists
INSTANCE_EXISTS=$(aws servicediscovery list-instances --service-id $APP_SERVICE_ID
 --query "Instances[?Id=='read-instance'].Id" --output text)

if [[ -z "$INSTANCE_EXISTS" ]]; then
  log_cmd "aws servicediscovery register-instance --service-id $APP_SERVICE_ID --
instance-id read-instance --attributes action=read,functionname=readfunction"
  OPERATION_ID=$(aws servicediscovery register-instance --service-id $APP_SERVICE_ID
 --instance-id read-instance --attributes action=read,functionname=readfunction --
query 'OperationId' --output text)

  # Wait for read instance registration to complete
  echo "Waiting for read instance registration to complete..." | tee -a $LOG_FILE
  wait_for_operation $OPERATION_ID
else
```

```
  echo "Instance read-instance already exists" | tee -a $LOG_FILE
fi

CREATED_RESOURCES+=("instance:$APP_SERVICE_ID:read-instance")

# Step 10: Create Python clients to interact with the services
echo "Step 10: Creating Python clients..." | tee -a $LOG_FILE

cat > writeclient.py << EOF
import boto3

serviceclient = boto3.client('servicediscovery')

response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
 ServiceName='app-service', QueryParameters={ 'action': 'write' })

functionname = response["Instances"][0]["Attributes"]["functionname"]

lambdaclient = boto3.client('lambda')

resp = lambdaclient.invoke(FunctionName=functionname, Payload='"This is a test
 data"')

print(resp["Payload"].read())
EOF

cat > readclient.py << EOF
import boto3

serviceclient = boto3.client('servicediscovery')

response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
 ServiceName='app-service', QueryParameters={ 'action': 'read' })

functionname = response["Instances"][0]["Attributes"]["functionname"]

lambdaclient = boto3.client('lambda')

resp = lambdaclient.invoke(FunctionName=functionname,
 InvocationType='RequestResponse')

print(resp["Payload"].read())
EOF
```

```
echo "Running write client..." | tee -a $LOG_FILE
log_cmd "python3 writeclient.py"

echo "Running read client..." | tee -a $LOG_FILE
log_cmd "python3 readclient.py"

# Step 11: Clean up resources
echo "Resources created:" | tee -a $LOG_FILE
for resource in "${CREATED_RESOURCES[@]}"; do
  echo "- $resource" | tee -a $LOG_FILE
done

echo "" | tee -a $LOG_FILE
echo "========================================" | tee -a $LOG_FILE
echo "CLEANUP CONFIRMATION" | tee -a $LOG_FILE
echo "========================================" | tee -a $LOG_FILE
echo "Do you want to clean up all created resources? (y/n): " | tee -a $LOG_FILE
read -r CLEANUP_CONFIRM
if [[ $CLEANUP_CONFIRM == "y" || $CLEANUP_CONFIRM == "Y" ]]; then
  cleanup
else
  echo "Resources were not cleaned up. You can manually clean them up later." | tee
 -a $LOG_FILE
fi

echo "Script completed at $(date)" | tee -a $LOG_FILE
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateHttpNamespace
  - CreateService
  - DeleteNamespace
  - DeleteService
  - DeregisterInstance
  - GetOperation
  - ListNamespaces
  - ListServices
  - RegisterInstance

Découverte du service Cloud Map

L'exemple de code suivant illustre comment :

- Créez un espace de noms DNS public avec l'intégration de la zone hébergée Route 53
- Créez des services détectables via des requêtes DNS et des appels d'API
- Enregistrez des instances de service avec différentes configurations de découverte
- Découvrez les services à l'aide de requêtes DNS et de l'API Cloud Map
- Vérifiez la découverte des services à l'aide de la commande dig et des appels d'API
- Nettoyez les ressources dans le bon ordre (instances, services, espace de noms)

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# AWS Cloud Map Private Namespace Tutorial Script
# This script demonstrates how to use AWS Cloud Map for service discovery
# with DNS queries and API calls

# Exit on error
set -e

# Configuration
REGION="us-east-2"
NAMESPACE_NAME="cloudmap-tutorial.com"
LOG_FILE="cloudmap-tutorial.log"
CREATOR_REQUEST_ID=$(date +%s)

# Function to log messages
log() {
    local message="$1"
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $message" | tee -a "$LOG_FILE"
}
```

```
# Function to check operation status
check_operation() {
    local operation_id="$1"
    local status=""

    log "Checking operation status for $operation_id..."

    while [[ "$status" != "SUCCESS" ]]; do
        sleep 5
        status=$(aws servicediscovery get-operation \
            --operation-id "$operation_id" \
            --region "$REGION" \
            --query "Operation.Status" \
            --output text)

        log "Operation status: $status"

        if [[ "$status" == "FAIL" ]]; then
            log "Operation failed. Exiting."
            exit 1
        fi
    done

    log "Operation completed successfully."
}

# Function to clean up resources
cleanup() {
    log "Starting cleanup process..."

    if [[ -n "$FIRST_INSTANCE_ID" ]]; then
        log "Deregistering first service instance..."
        aws servicediscovery deregister-instance \
            --service-id "$PUBLIC_SERVICE_ID" \
            --instance-id "$FIRST_INSTANCE_ID" \
            --region "$REGION" || log "Failed to deregister first instance"
    fi

    if [[ -n "$SECOND_INSTANCE_ID" ]]; then
        log "Deregistering second service instance..."
        aws servicediscovery deregister-instance \
            --service-id "$BACKEND_SERVICE_ID" \
            --instance-id "$SECOND_INSTANCE_ID" \
            --region "$REGION" || log "Failed to deregister second instance"
```

```
    fi

    if [[ -n "$PUBLIC_SERVICE_ID" ]]; then
        log "Deleting public service..."
        aws servicediscovery delete-service \
            --id "$PUBLIC_SERVICE_ID" \
            --region "$REGION" || log "Failed to delete public service"
    fi

    if [[ -n "$BACKEND_SERVICE_ID" ]]; then
        log "Deleting backend service..."
        aws servicediscovery delete-service \
            --id "$BACKEND_SERVICE_ID" \
            --region "$REGION" || log "Failed to delete backend service"
    fi

    if [[ -n "$NAMESPACE_ID" ]]; then
        log "Deleting namespace..."
        aws servicediscovery delete-namespace \
            --id "$NAMESPACE_ID" \
            --region "$REGION" || log "Failed to delete namespace"
    fi

    log "Cleanup completed."
}

# Set up trap for cleanup on script exit
trap cleanup EXIT INT TERM

# Initialize log file
> "$LOG_FILE"
log "Starting AWS Cloud Map tutorial script"

# Step 1: Create an AWS Cloud Map namespace
log "Creating AWS Cloud Map namespace: $NAMESPACE_NAME"
OPERATION_RESULT=$(aws servicediscovery create-public-dns-namespace \
    --name "$NAMESPACE_NAME" \
    --creator-request-id "cloudmap-tutorial-$CREATOR_REQUEST_ID" \
    --region "$REGION")

OPERATION_ID=$(echo "$OPERATION_RESULT" | jq -r '.OperationId')
log "Namespace creation initiated. Operation ID: $OPERATION_ID"

# Check operation status
```

```
check_operation "$OPERATION_ID"

# Get the namespace ID
log "Getting namespace ID..."
NAMESPACE_ID=$(aws servicediscovery list-namespaces \
    --region "$REGION" \
    --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" \
    --output text)

log "Namespace ID: $NAMESPACE_ID"

# Get the hosted zone ID
log "Getting Route 53 hosted zone ID..."
HOSTED_ZONE_ID=$(aws route53 list-hosted-zones-by-name \
    --dns-name "$NAMESPACE_NAME" \
    --query "HostedZones[0].Id" \
    --output text | sed 's|/hostedzone/||')

log "Hosted Zone ID: $HOSTED_ZONE_ID"

# Step 2: Create the AWS Cloud Map services
log "Creating public service..."
PUBLIC_SERVICE_RESULT=$(aws servicediscovery create-service \
    --name "public-service" \
    --namespace-id "$NAMESPACE_ID" \
    --dns-config "RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=300}]" \
    --region "$REGION")

PUBLIC_SERVICE_ID=$(echo "$PUBLIC_SERVICE_RESULT" | jq -r '.Service.Id')
log "Public service created. Service ID: $PUBLIC_SERVICE_ID"

log "Creating backend service..."
BACKEND_SERVICE_RESULT=$(aws servicediscovery create-service \
    --name "backend-service" \
    --namespace-id "$NAMESPACE_ID" \
    --type "HTTP" \
    --region "$REGION")

BACKEND_SERVICE_ID=$(echo "$BACKEND_SERVICE_RESULT" | jq -r '.Service.Id')
log "Backend service created. Service ID: $BACKEND_SERVICE_ID"

# Step 3: Register the AWS Cloud Map service instances
log "Registering first service instance..."
FIRST_INSTANCE_RESULT=$(aws servicediscovery register-instance \
```

```
    --service-id "$PUBLIC_SERVICE_ID" \
    --instance-id "first" \
    --attributes "AWS_INSTANCE_IPV4=192.168.2.1" \
    --region "$REGION")

FIRST_INSTANCE_ID="first"
FIRST_OPERATION_ID=$(echo "$FIRST_INSTANCE_RESULT" | jq -r '.OperationId')
log "First instance registration initiated. Operation ID: $FIRST_OPERATION_ID"

# Check operation status
check_operation "$FIRST_OPERATION_ID"

log "Registering second service instance..."
SECOND_INSTANCE_RESULT=$(aws servicediscovery register-instance \
    --service-id "$BACKEND_SERVICE_ID" \
    --instance-id "second" \
    --attributes "service-name=backend" \
    --region "$REGION")

SECOND_INSTANCE_ID="second"
SECOND_OPERATION_ID=$(echo "$SECOND_INSTANCE_RESULT" | jq -r '.OperationId')
log "Second instance registration initiated. Operation ID: $SECOND_OPERATION_ID"

# Check operation status
check_operation "$SECOND_OPERATION_ID"

# Step 4: Discover the AWS Cloud Map service instances
log "Getting Route 53 name servers..."
NAME_SERVERS=$(aws route53 get-hosted-zone \
    --id "$HOSTED_ZONE_ID" \
    --query "DelegationSet.NameServers[0]" \
    --output text)

log "Name server: $NAME_SERVERS"

log "Using dig to query DNS records (this will be simulated)..."
log "Command: dig @$NAME_SERVERS public-service.$NAMESPACE_NAME"
log "Expected output would show: public-service.$NAMESPACE_NAME. 300 IN A
 192.168.2.1"

log "Using AWS CLI to discover backend service instances..."
DISCOVER_RESULT=$(aws servicediscovery discover-instances \
    --namespace-name "$NAMESPACE_NAME" \
    --service-name "backend-service" \
```

```
    --region "$REGION")

log "Discovery result: $(echo "$DISCOVER_RESULT" | jq -c '.')"

# Display created resources
log "Resources created:"
log "- Namespace: $NAMESPACE_NAME (ID: $NAMESPACE_ID)"
log "- Public Service: public-service (ID: $PUBLIC_SERVICE_ID)"
log "- Backend Service: backend-service (ID: $BACKEND_SERVICE_ID)"
log "- Service Instance: first (Service: public-service)"
log "- Service Instance: second (Service: backend-service)"

# Ask user if they want to clean up resources
read -p "Do you want to clean up all created resources? (y/n): " CLEANUP_RESPONSE

if [[ "$CLEANUP_RESPONSE" == "y" || "$CLEANUP_RESPONSE" == "Y" ]]; then
    log "User confirmed cleanup. Proceeding with resource deletion."
    # Cleanup function will be called automatically on exit
else
    log "User chose not to clean up resources. Exiting without cleanup."
    trap - EXIT
    exit 0
fi
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreatePublicDnsNamespace
  - CreateService
  - DeleteNamespace
  - DeleteService
  - DeregisterInstance
  - DiscoverInstances
  - GetOperation
  - ListNamespaces
  - RegisterInstance

CloudFront exemples d'utilisation AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with CloudFront.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Scénarios

Scénarios

Commencez avec CloudFront

L'exemple de code suivant illustre comment :

- Création d'un compartiment Amazon S3 pour le stockage de contenu
- Chargez un exemple de contenu dans le compartiment S3
- Créez un contrôle d'accès à l'origine (OAC) pour un accès S3 sécurisé
- Création d'une CloudFront distribution avec S3 comme origine
- Mettre à jour la politique du compartiment S3 pour autoriser CloudFront l'accès
- Attendre le déploiement de la distribution et tester l'accès au contenu
- Nettoyez les ressources, y compris la distribution, l'OAC et le compartiment S3

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# CloudFront Getting Started Tutorial Script
# This script creates an S3 bucket, uploads sample content, creates a CloudFront
 distribution with OAC,
# and demonstrates how to access content through CloudFront.

# Set up logging
LOG_FILE="cloudfront-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting CloudFront Getting Started Tutorial at $(date)"

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Resources created before error:"
    if [ -n "$BUCKET_NAME" ]; then
        echo "- S3 Bucket: $BUCKET_NAME"
    fi
    if [ -n "$OAC_ID" ]; then
        echo "- CloudFront Origin Access Control: $OAC_ID"
    fi
    if [ -n "$DISTRIBUTION_ID" ]; then
        echo "- CloudFront Distribution: $DISTRIBUTION_ID"
    fi

    echo "Attempting to clean up resources..."
    cleanup
    exit 1
}

# Function to clean up resources
cleanup() {
    echo "Cleaning up resources..."

    if [ -n "$DISTRIBUTION_ID" ]; then
        echo "Disabling CloudFront distribution $DISTRIBUTION_ID..."

        # Get the current configuration and ETag
        ETAG=$(aws cloudfront get-distribution-config --id "$DISTRIBUTION_ID" --
query 'ETag' --output text)
        if [ $? -ne 0 ]; then
```

```
            echo "Failed to get distribution config. Continuing with cleanup..."
        else
            # Create a modified configuration with Enabled=false
            aws cloudfront get-distribution-config --id "$DISTRIBUTION_ID" | \
            jq '.DistributionConfig.Enabled = false' > temp_disabled_config.json

            # Update the distribution to disable it
            aws cloudfront update-distribution \
                --id "$DISTRIBUTION_ID" \
                --distribution-config file://<(jq '.DistributionConfig'
 temp_disabled_config.json) \
                --if-match "$ETAG"

            if [ $? -ne 0 ]; then
                echo "Failed to disable distribution. Continuing with cleanup..."
            else
                echo "Waiting for distribution to be disabled (this may take several
 minutes)..."
                aws cloudfront wait distribution-deployed --id "$DISTRIBUTION_ID"

                # Delete the distribution
                ETAG=$(aws cloudfront get-distribution-config --id
 "$DISTRIBUTION_ID" --query 'ETag' --output text)
                aws cloudfront delete-distribution --id "$DISTRIBUTION_ID" --if-
match "$ETAG"
                if [ $? -ne 0 ]; then
                    echo "Failed to delete distribution. You may need to delete it
 manually."
                else
                    echo "CloudFront distribution deleted."
                fi
            fi
        fi
    fi

    if [ -n "$OAC_ID" ]; then
        echo "Deleting Origin Access Control $OAC_ID..."
        OAC_ETAG=$(aws cloudfront get-origin-access-control --id "$OAC_ID" --query
 'ETag' --output text 2>/dev/null)
        if [ $? -ne 0 ]; then
            echo "Failed to get Origin Access Control ETag. You may need to delete
 it manually."
        else
```

```
            aws cloudfront delete-origin-access-control --id "$OAC_ID" --if-match
 "$OAC_ETAG"
            if [ $? -ne 0 ]; then
                echo "Failed to delete Origin Access Control. You may need to delete
 it manually."
            else
                echo "Origin Access Control deleted."
            fi
        fi
    fi

    if [ -n "$BUCKET_NAME" ]; then
        echo "Deleting S3 bucket $BUCKET_NAME and its contents..."
        aws s3 rm "s3://$BUCKET_NAME" --recursive
        if [ $? -ne 0 ]; then
            echo "Failed to remove bucket contents. Continuing with bucket
 deletion..."
        fi

        aws s3 rb "s3://$BUCKET_NAME"
        if [ $? -ne 0 ]; then
            echo "Failed to delete bucket. You may need to delete it manually."
        else
            echo "S3 bucket deleted."
        fi
    fi

    # Clean up temporary files
    rm -f temp_disabled_config.json
    rm -rf temp_content
}

# Generate a random identifier for the bucket name
RANDOM_ID=$(openssl rand -hex 6)
BUCKET_NAME="cloudfront-${RANDOM_ID}"
echo "Using bucket name: $BUCKET_NAME"

# Create a temporary directory for content
TEMP_DIR="temp_content"
mkdir -p "$TEMP_DIR/css"
if [ $? -ne 0 ]; then
    handle_error "Failed to create temporary directory"
fi
```

```
# Step 1: Create an S3 bucket
echo "Creating S3 bucket: $BUCKET_NAME"
aws s3 mb "s3://$BUCKET_NAME"
if [ $? -ne 0 ]; then
    handle_error "Failed to create S3 bucket"
fi

# Step 2: Create sample content
echo "Creating sample content..."
cat > "$TEMP_DIR/index.html" << 'EOF'
<!DOCTYPE html>
<html>
<head>
    <title>Hello World</title>
    <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<body>
    <h1>Hello world!</h1>
</body>
</html>
EOF

cat > "$TEMP_DIR/css/styles.css" << 'EOF'
body {
    font-family: Arial, sans-serif;
    margin: 40px;
    background-color: #f5f5f5;
}
h1 {
    color: #333;
    text-align: center;
}
EOF

# Step 3: Upload content to the S3 bucket
echo "Uploading content to S3 bucket..."
aws s3 cp "$TEMP_DIR/" "s3://$BUCKET_NAME/" --recursive
if [ $? -ne 0 ]; then
    handle_error "Failed to upload content to S3 bucket"
fi

# Step 4: Create Origin Access Control
echo "Creating Origin Access Control..."
OAC_RESPONSE=$(aws cloudfront create-origin-access-control \
```

```
    --origin-access-control-config Name="oac-for-
$BUCKET_NAME",SigningProtocol=sigv4,SigningBehavior=always,OriginAccessControlOriginType=s3)

if [ $? -ne 0 ]; then
    handle_error "Failed to create Origin Access Control"
fi

OAC_ID=$(echo "$OAC_RESPONSE" | jq -r '.OriginAccessControl.Id')
echo "Created Origin Access Control with ID: $OAC_ID"

# Step 5: Create CloudFront distribution
echo "Creating CloudFront distribution..."

# Get AWS account ID for bucket policy
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
if [ $? -ne 0 ]; then
    handle_error "Failed to get AWS account ID"
fi

# Create distribution configuration
cat > distribution-config.json << EOF
{
    "CallerReference": "cli-tutorial-$(date +%s)",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "S3-$BUCKET_NAME",
                "DomainName": "$BUCKET_NAME.s3.amazonaws.com",
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                },
                "OriginAccessControlId": "$OAC_ID"
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "S3-$BUCKET_NAME",
        "ViewerProtocolPolicy": "redirect-to-https",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["GET", "HEAD"],
            "CachedMethods": {
                "Quantity": 2,
```

```
                "Items": ["GET", "HEAD"]
            }
        },
        "DefaultTTL": 86400,
        "MinTTL": 0,
        "MaxTTL": 31536000,
        "Compress": true,
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            }
        }
    },
    "Comment": "CloudFront distribution for tutorial",
    "Enabled": true,
    "WebACLId": ""
}
EOF

DIST_RESPONSE=$(aws cloudfront create-distribution --distribution-config file://
distribution-config.json)
if [ $? -ne 0 ]; then
    handle_error "Failed to create CloudFront distribution"
fi

DISTRIBUTION_ID=$(echo "$DIST_RESPONSE" | jq -r '.Distribution.Id')
DOMAIN_NAME=$(echo "$DIST_RESPONSE" | jq -r '.Distribution.DomainName')

echo "Created CloudFront distribution with ID: $DISTRIBUTION_ID"
echo "CloudFront domain name: $DOMAIN_NAME"

# Step 6: Update S3 bucket policy
echo "Updating S3 bucket policy..."
cat > bucket-policy.json << EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
```

```
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::$BUCKET_NAME/*",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceArn": "arn:aws:cloudfront::$ACCOUNT_ID:distribution/
$DISTRIBUTION_ID"
                }
            }
        }
    ]
}
EOF

aws s3api put-bucket-policy --bucket "$BUCKET_NAME" --policy file://bucket-
policy.json
if [ $? -ne 0 ]; then
    handle_error "Failed to update S3 bucket policy"
fi

# Step 7: Wait for distribution to deploy
echo "Waiting for CloudFront distribution to deploy (this may take 5-10 minutes)..."
aws cloudfront wait distribution-deployed --id "$DISTRIBUTION_ID"
if [ $? -ne 0 ]; then
    echo "Warning: Distribution deployment wait timed out. The distribution may
 still be deploying."
else
    echo "CloudFront distribution is now deployed."
fi

# Step 8: Display access information
echo ""
echo "===== CloudFront Distribution Setup Complete ====="
echo "You can access your content at: https://$DOMAIN_NAME/index.html"
echo ""
echo "Resources created:"
echo "- S3 Bucket: $BUCKET_NAME"
echo "- CloudFront Origin Access Control: $OAC_ID"
echo "- CloudFront Distribution: $DISTRIBUTION_ID"
echo ""

# Ask user if they want to clean up resources
read -p "Do you want to clean up all resources created by this script? (y/n): "
 CLEANUP_RESPONSE
if [[ "$CLEANUP_RESPONSE" =~ ^[Yy] ]]; then
```

```
    cleanup
    echo "All resources have been cleaned up."
else
    echo "Resources will not be cleaned up. You can manually delete them later."
    echo "To access your content, visit: https://$DOMAIN_NAME/index.html"
fi

echo "Tutorial completed at $(date)"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateDistribution
  - CreateOriginAccessControl
  - DeleteDistribution
  - DeleteOriginAccessControl
  - GetDistribution
  - GetDistributionConfig
  - GetOriginAccessControl
  - UpdateDistribution
  - WaitDistributionDeployed

Exemples AWS CLI d'utilisation de DynamoDB avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash avec DynamoDB.

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Principes de base
- Actions
- Scénarios

Principes de base

Principes de base

L'exemple de code suivant illustre comment :

- Créez une table pouvant contenir des données vidéo.
- Insérer, récupérez et mettez à jour un seul film dans la table.
- Écrivez des données vidéo dans la table à partir d'un exemple de fichier JSON.
- Recherchez les films sortis au cours d'une année donnée.
- Recherchez les films sortis au cours d'une plage d'années spécifique.
- Supprimez un film de la table, puis supprimez la table.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Scénario de démarrage de DynamoDB.

```
###############################################################################
# function dynamodb_getting_started_movies
#
# Scenario to create an Amazon DynamoDB table and perform a series of operations on
 the table.
#
```

```
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function dynamodb_getting_started_movies() {

  source ./dynamodb_operations.sh

  key_schema_json_file="dynamodb_key_schema.json"
  attribute_definitions_json_file="dynamodb_attr_def.json"
  item_json_file="movie_item.json"
  key_json_file="movie_key.json"
  batch_json_file="batch.json"
  attribute_names_json_file="attribute_names.json"
  attributes_values_json_file="attribute_values.json"

  echo_repeat "*" 88
  echo
  echo "Welcome to the Amazon DynamoDB getting started demo."
  echo
  echo_repeat "*" 88
  echo

  local table_name
  echo -n "Enter a name for a new DynamoDB table: "
  get_input
  table_name=$get_input_result

  echo '[
  {"AttributeName": "year", "KeyType": "HASH"},
   {"AttributeName": "title", "KeyType": "RANGE"}
  ]' >"$key_schema_json_file"

  echo '[
  {"AttributeName": "year", "AttributeType": "N"},
   {"AttributeName": "title", "AttributeType": "S"}
  ]' >"$attribute_definitions_json_file"

  if dynamodb_create_table -n "$table_name" -a "$attribute_definitions_json_file" \
    -k "$key_schema_json_file" 1>/dev/null; then
    echo "Created a DynamoDB table named $table_name"
  else
    errecho "The table failed to create. This demo will exit."
    clean_up
```

```
    return 1
  fi

  echo "Waiting for the table to become active...."

  if dynamodb_wait_table_active -n "$table_name"; then
    echo "The table is now active."
  else
    errecho "The table failed to become active. This demo will exit."
    cleanup "$table_name"
    return 1
  fi

  echo
  echo_repeat "*" 88
  echo

  echo -n "Enter the title of a movie you want to add to the table: "
  get_input
  local added_title
  added_title=$get_input_result

  local added_year
  get_int_input "What year was it released? "
  added_year=$get_input_result

  local rating
  get_float_input "On a scale of 1 - 10, how do you rate it? " "1" "10"
  rating=$get_input_result

  local plot
  echo -n "Summarize the plot for me: "
  get_input
  plot=$get_input_result

  echo '{
    "year": {"N" :"'"$added_year"'"},
    "title": {"S" :  "'"$added_title"'"},
    "info": {"M" : {"plot": {"S" : "'"$plot"'"}, "rating": {"N" :"'"$rating"'"} } }
   }' >"$item_json_file"

  if dynamodb_put_item -n "$table_name" -i "$item_json_file"; then
    echo "The movie '$added_title' was successfully added to the table
'$table_name'."
```

```
  else
    errecho "Put item failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi

  echo
  echo_repeat "*" 88
  echo

  echo "Let's update your movie '$added_title'."
  get_float_input "You rated it $rating, what new rating would you give it? " "1"
"10"
  rating=$get_input_result

  echo -n "You summarized the plot as '$plot'."
  echo "What would you say now? "
  get_input
  plot=$get_input_result

  echo '{
    "year": {"N" :"'"$added_year"'"},
    "title": {"S" :  "'"$added_title"'"}
    }' >"$key_json_file"

  echo '{
    ":r": {"N" :"'"$rating"'"},
    ":p": {"S" : "'"$plot"'"}
   }' >"$item_json_file"

  local update_expression="SET info.rating = :r, info.plot = :p"

  if dynamodb_update_item -n "$table_name" -k "$key_json_file" -e
"$update_expression" -v "$item_json_file"; then
    echo "Updated '$added_title' with new attributes."
  else
    errecho "Update item failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi

  echo
  echo_repeat "*" 88
  echo
```

```
  echo "We will now use batch write to upload 150 movie entries into the table."

  local batch_json
  for batch_json in movie_files/movies_*.json; do
    echo "{ \"$table_name\" : $(<"$batch_json") }" >"$batch_json_file"
    if dynamodb_batch_write_item -i "$batch_json_file" 1>/dev/null; then
      echo "Entries in $batch_json added to table."
    else
      errecho "Batch write failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi
  done

  local title="The Lord of the Rings: The Fellowship of the Ring"
  local year="2001"

  if get_yes_no_input "Let's move on...do you want to get info about '$title'? (y/n)
"; then
    echo '{
  "year": {"N" :"'"$year"'"},
  "title": {"S" :  "'"$title"'"}
  }' >"$key_json_file"
    local info
    info=$(dynamodb_get_item -n "$table_name" -k "$key_json_file")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
      errecho "Get item failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi

    echo "Here is what I found:"
    echo "$info"
  fi

  local ask_for_year=true
  while [[ "$ask_for_year" == true ]]; do
    echo "Let's get a list of movies released in a given year."
    get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"
    year=$get_input_result
    echo '{
```

```
    "#n": "year"
    }' >"$attribute_names_json_file"

    echo '{
    ":v": {"N" :"'"$year"'"}
    }' >"$attributes_values_json_file"

    response=$(dynamodb_query -n "$table_name" -k "#n=:v" -a
"$attribute_names_json_file" -v "$attributes_values_json_file")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
      errecho "Query table failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi

    echo "Here is what I found:"
    echo "$response"

    if ! get_yes_no_input "Try another year? (y/n) "; then
      ask_for_year=false
    fi
  done

  echo "Now let's scan for movies released in a range of years. Enter a year: "
  get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"
  local start=$get_input_result

  get_int_input "Enter another year: " "1972" "2018"
  local end=$get_input_result

  echo '{
    "#n": "year"
    }' >"$attribute_names_json_file"

  echo '{
    ":v1": {"N" : "'"$start"'"},
    ":v2": {"N" : "'"$end"'"}
    }' >"$attributes_values_json_file"

  response=$(dynamodb_scan -n "$table_name" -f "#n BETWEEN :v1 AND :v2" -a
"$attribute_names_json_file" -v "$attributes_values_json_file")
```

```
  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "Scan table failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi

  echo "Here is what I found:"
  echo "$response"

  echo
  echo_repeat "*" 88
  echo

  echo "Let's remove your movie '$added_title' from the table."

  if get_yes_no_input "Do you want to remove '$added_title'? (y/n) "; then
    echo '{
  "year": {"N" :"'"$added_year"'"},
  "title": {"S" :  "'"$added_title"'"}
  }' >"$key_json_file"

    if ! dynamodb_delete_item -n "$table_name" -k "$key_json_file"; then
      errecho "Delete item failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi
  fi

  if get_yes_no_input "Do you want to delete the table '$table_name'? (y/n) "; then
    if ! clean_up "$table_name"; then
      return 1
    fi
  else
    if ! clean_up; then
      return 1
    fi
  fi

  return 0
}
```

Fonctions DynamoDB utilisées dans ce scénario.

```
###############################################################################
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to create.
#       -a attribute_definitions -- JSON file path of a list of attributes and their
 types.
#       -k key_schema -- JSON file path of a list of attributes and their key types.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_create_table() {
  local table_name attribute_definitions key_schema response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table with on-demand billing."
    echo " -n table_name  -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
 their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
 types."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:a:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      a) attribute_definitions="${OPTARG}" ;;
      k) key_schema="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$attribute_definitions" ]]; then
  errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
  usage
  return 1
fi

if [[ -z "$key_schema" ]]; then
  errecho "ERROR: You must provide a key schema json file path the -k parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    table_name:   $table_name"
iecho "    attribute_definitions:   $attribute_definitions"
iecho "    key_schema:   $key_schema"
iecho ""

response=$(aws dynamodb create-table \
  --table-name "$table_name" \
  --attribute-definitions file://"$attribute_definitions" \
  --billing-mode PAY_PER_REQUEST \
  --key-schema file://"$key_schema" )

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
  fi

  return 0
}

###############################################################################
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#
#  Response:
#       - TableStatus:
#     And:
#       0 - Table is active.
#       1 - If it fails.
###############################################################################
function dynamodb_describe_table {
  local table_name
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_describe_table"
    echo "Describe the status of a DynamoDB table."
    echo "  -n table_name  -- The name of the table."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  local table_status
    table_status=$(
      aws dynamodb describe-table \
        --table-name "$table_name" \
        --output text \
        --query 'Table.TableStatus'
    )

   local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log "$error_code"
      errecho "ERROR: AWS reports describe-table operation failed.$table_status"
      return 1
    fi

  echo "$table_status"

  return 0
}

###############################################################################
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -i item  -- Path to json file containing the item values.
```

```
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_put_item() {
  local table_name item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -i item  -- Path to json file containing the item values."
    echo ""
  }

  while getopts "n:i:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$item" ]]; then
```

```
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    item:   $item"
  iecho ""
  iecho ""

  response=$(aws dynamodb put-item \
    --table-name "$table_name" \
    --item file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.$response"
    return 1
  fi

  return 0

}

###############################################################################
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 update.
#       -e update expression  -- An expression that defines one or more attributes
 to be updated.
#       -v values  -- Path to json file containing the update values.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
```

```
###############################################################################
function dynamodb_update_item() {
  local table_name keys update_expression values response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_update_item"
    echo "Update an item in a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to update."
    echo " -e update expression  -- An expression that defines one or more
 attributes to be updated."
    echo " -v values  -- Path to json file containing the update values."
    echo ""
  }

  while getopts "n:k:e:v:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      e) update_expression="${OPTARG}" ;;
      v) values="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi
```

```
  if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
  fi
  if [[ -z "$update_expression" ]]; then
    errecho "ERROR: You must provide an update expression with the -e parameter."
    usage
    return 1
  fi

  if [[ -z "$values" ]]; then
    errecho "ERROR: You must provide a values json file path the -v parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    keys:   $keys"
  iecho "    update_expression:   $update_expression"
  iecho "    values:   $values"

  response=$(aws dynamodb update-item \
    --table-name "$table_name" \
    --key file://"$keys" \
    --update-expression "$update_expression" \
    --expression-attribute-values file://"$values")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
  fi

  return 0

}

################################################################################
# function dynamodb_batch_write_item
```

```
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#       -i item  -- Path to json file containing the items to write.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_batch_write_item() {
  local item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_batch_write_item"
    echo "Write a batch of items into a DynamoDB table."
    echo " -i item  -- Path to json file containing the items to write."
    echo ""
  }
  while getopts "i:h" option; do
    case "${option}" in
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi
```

```
  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    item:   $item"
  iecho ""

  response=$(aws dynamodb batch-write-item \
    --request-items file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
  fi

  return 0
}

###############################################################################
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 get.
#       [-q query]  -- Optional JMESPath query expression.
#
#  Returns:
#       The item as text output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_get_item() {
  local table_name keys query response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
```

```
 function usage() {
   echo "function dynamodb_get_item"
   echo "Get an item from a DynamoDB table."
   echo " -n table_name  -- The name of the table."
   echo " -k keys  -- Path to json file containing the keys that identify the item
to get."
   echo " [-q query]  -- Optional JMESPath query expression."
   echo ""
 }
 query=""
 while getopts "n:k:q:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     k) keys="${OPTARG}" ;;
     q) query="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$keys" ]]; then
   errecho "ERROR: You must provide a keys json file path the -k parameter."
   usage
   return 1
 fi

 if [[ -n "$query" ]]; then
   response=$(aws dynamodb get-item \
     --table-name "$table_name" \
     --key file://"$keys" \
```

```
      --output text \
      --query "$query")
  else
    response=$(
      aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"$keys" \
        --output text
    )
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
  fi

  if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
 query inserts on some strings.
  else
    echo "$response"
  fi

  return 0
}

############################################################################
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k key_condition_expression -- The key condition expression.
#       -a attribute_names -- Path to JSON file containing the attribute names.
#       -v attribute_values -- Path to JSON file containing the attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
```

```
#       0 - If successful.
#       1 - If it fails.
################################################################################
function dynamodb_query() {
  local table_name key_condition_expression attribute_names attribute_values
 projection_expression response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_query"
    echo "Query a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k key_condition_expression -- The key condition expression."
    echo " -a attribute_names -- Path to JSON file containing the attribute names."
    echo " -v attribute_values -- Path to JSON file containing the attribute
 values."
    echo " [-p projection_expression]  -- Optional projection expression."
    echo ""
  }

  while getopts "n:k:a:v:p:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) key_condition_expression="${OPTARG}" ;;
      a) attribute_names="${OPTARG}" ;;
      v) attribute_values="${OPTARG}" ;;
      p) projection_expression="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
```

```
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$key_condition_expression" ]]; then
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
  fi

  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values")
  else
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
```

```
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function dynamodb_scan
#
# This function scans a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -f filter_expression  -- The filter expression.
#       -a expression_attribute_names -- Path to JSON file containing the expression
 attribute names.
#       -v expression_attribute_values -- Path to JSON file containing the
 expression attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_scan() {
  local table_name filter_expression expression_attribute_names
 expression_attribute_values projection_expression response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_scan"
    echo "Scan a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -f filter_expression  -- The filter expression."
    echo " -a expression_attribute_names -- Path to JSON file containing the
 expression attribute names."
```

```
   echo " -v expression_attribute_values -- Path to JSON file containing the
expression attribute values."
   echo " [-p projection_expression]  -- Optional projection expression."
   echo ""
 }

 while getopts "n:f:a:v:p:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     f) filter_expression="${OPTARG}" ;;
     a) expression_attribute_names="${OPTARG}" ;;
     v) expression_attribute_values="${OPTARG}" ;;
     p) projection_expression="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$filter_expression" ]]; then
   errecho "ERROR: You must provide a filter expression with the -f parameter."
   usage
   return 1
 fi

 if [[ -z "$expression_attribute_names" ]]; then
   errecho "ERROR: You must provide expression attribute names with the -a
parameter."
   usage
   return 1
 fi
```

```
  if [[ -z "$expression_attribute_values" ]]; then
    errecho "ERROR: You must provide expression attribute values with the -v
 parameter."
    usage
    return 1
  fi

  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb scan \
      --table-name "$table_name" \
      --filter-expression "$filter_expression" \
      --expression-attribute-names file://"$expression_attribute_names" \
      --expression-attribute-values file://"$expression_attribute_values")
  else
    response=$(aws dynamodb scan \
      --table-name "$table_name" \
      --filter-expression "$filter_expression" \
      --expression-attribute-names file://"$expression_attribute_names" \
      --expression-attribute-values file://"$expression_attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports scan operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
```

```
#       -k keys  -- Path to json file containing the keys that identify the item to
 delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_item() {
  local table_name keys response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to delete."
    echo ""
  }
  while getopts "n:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi
```

```
  if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    keys:   $keys"
  iecho ""

  response=$(aws dynamodb delete-item \
    --table-name "$table_name" \
    --key file://"$keys")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
  fi

  return 0

}

###############################################################################
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_table() {
  local table_name response
  local option OPTARG # Required to use getopts command in a function.
```

```
# bashsupport disable=BP5008
function usage() {
  echo "function dynamodb_delete_table"
  echo "Deletes an Amazon DynamoDB table."
  echo " -n table_name  -- The name of the table to delete."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) table_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    table_name:   $table_name"
iecho ""

response=$(aws dynamodb delete-table \
  --table-name "$table_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-table operation failed.$response"
  return 1
```

```
  fi

  return 0
}
```

Fonctions utilitaires utilisées dans ce scénario.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
```

```
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - Interrogation
  - Analyser
  - UpdateItem

Actions

`BatchGetItem`

L'exemple de code suivant montre comment utiliser`BatchGetItem`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_batch_get_item
#
# This function gets a batch of items from a DynamoDB table.
#
# Parameters:
#       -i item  -- Path to json file containing the keys of the items to get.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_batch_get_item() {
  local item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_batch_get_item"
    echo "Get a batch of items from a DynamoDB table."
    echo " -i item  -- Path to json file containing the keys of the items to get."
    echo ""
  }
```

```
  while getopts "i:h" option; do
    case "${option}" in
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  response=$(aws dynamodb batch-get-item \
    --request-items file://"$item")
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-get-item operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
```

```
# This function outputs everything sent to it to STDERR (standard error output).
#########################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#########################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#########################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous BatchGetItemà la section Référence des AWS CLI commandes.

`BatchWriteItem`

L'exemple de code suivant montre comment utiliser`BatchWriteItem`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#       -i item  -- Path to json file containing the items to write.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_batch_write_item() {
  local item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_batch_write_item"
    echo "Write a batch of items into a DynamoDB table."
    echo " -i item  -- Path to json file containing the items to write."
    echo ""
  }
  while getopts "i:h" option; do
    case "${option}" in
```

```
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    item:   $item"
  iecho ""

  response=$(aws dynamodb batch-write-item \
    --request-items file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
  fi

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
################################################################################
```

```
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
```

```
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous BatchWriteItemà la section Référence des AWS CLI commandes.

`CreateTable`

L'exemple de code suivant montre comment utiliser`CreateTable`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to create.
#       -a attribute_definitions -- JSON file path of a list of attributes and their
 types.
#       -k key_schema -- JSON file path of a list of attributes and their key types.
#
#  Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_create_table() {
  local table_name attribute_definitions key_schema response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table with on-demand billing."
    echo " -n table_name  -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
 their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
 types."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:a:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      a) attribute_definitions="${OPTARG}" ;;
      k) key_schema="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
```

```
  fi

  if [[ -z "$attribute_definitions" ]]; then
    errecho "ERROR: You must provide an attribute definitions json file path the -a
 parameter."
    usage
    return 1
  fi

  if [[ -z "$key_schema" ]]; then
    errecho "ERROR: You must provide a key schema json file path the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --billing-mode PAY_PER_REQUEST \
    --key-schema file://"$key_schema" )

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
  fi

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function iecho
```

```
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
```

```
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateTableà la section Référence des AWS CLI commandes.

`DeleteItem`

L'exemple de code suivant montre comment utiliser`DeleteItem`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
```

```
###############################################################################
function dynamodb_delete_item() {
  local table_name keys response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to delete."
    echo ""
  }
  while getopts "n:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
  fi
```

```
  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    keys:   $keys"
  iecho ""

  response=$(aws dynamodb delete-item \
    --table-name "$table_name" \
    --key file://"$keys")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
  fi

  return 0

}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
```

```
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteItemà la section Référence des AWS CLI commandes.

`DeleteTable`

L'exemple de code suivant montre comment utiliser`DeleteTable`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_table() {
  local table_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function dynamodb_delete_table"
    echo "Deletes an Amazon DynamoDB table."
    echo " -n table_name  -- The name of the table to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho ""

  response=$(aws dynamodb delete-table \
    --table-name "$table_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-table operation failed.$response"
    return 1
  fi

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
```

```
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
```

```
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteTableà la section Référence des AWS CLI commandes.

`DescribeTable`

L'exemple de code suivant montre comment utiliser`DescribeTable`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#
#  Response:
#       - TableStatus:
#     And:
#       0 - Table is active.
#       1 - If it fails.
###############################################################################
function dynamodb_describe_table {
  local table_name
  local option OPTARG # Required to use getopts command in a function.
```

```
#####################################
# Function usage explanation
#####################################
function usage() {
  echo "function dynamodb_describe_table"
  echo "Describe the status of a DynamoDB table."
  echo "  -n table_name  -- The name of the table."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) table_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

local table_status
  table_status=$(
    aws dynamodb describe-table \
      --table-name "$table_name" \
      --output text \
      --query 'Table.TableStatus'
  )

 local error_code=${?}
```

```
    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log "$error_code"
      errecho "ERROR: AWS reports describe-table operation failed.$table_status"
      return 1
    fi

  echo "$table_status"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
```

```
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeTableà la section Référence des AWS CLI commandes.

`GetItem`

L'exemple de code suivant montre comment utiliser`GetItem`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
```

```
#       -k keys  -- Path to json file containing the keys that identify the item to
 get.
#       [-q query]  -- Optional JMESPath query expression.
#
#  Returns:
#       The item as text output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_get_item() {
  local table_name keys query response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_get_item"
    echo "Get an item from a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to get."
    echo " [-q query]  -- Optional JMESPath query expression."
    echo ""
  }
  query=""
  while getopts "n:k:q:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      q) query="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
  fi

  if [[ -n "$query" ]]; then
    response=$(aws dynamodb get-item \
      --table-name "$table_name" \
      --key file://"$keys" \
      --output text \
      --query "$query")
  else
    response=$(
      aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"$keys" \
        --output text
    )
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
  fi

  if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
query inserts on some strings.
  else
    echo "$response"
  fi

  return 0
```

```
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
```

```
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous GetItemà la section Référence des AWS CLI commandes.

`ListTables`

L'exemple de code suivant montre comment utiliser`ListTables`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_list_tables
#
# This function lists all the tables in a DynamoDB.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_list_tables() {
  response=$(aws dynamodb list-tables \
    --output text \
    --query "TableNames")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
```

```
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
  fi

  echo "$response" | tr -s "[:space:]" "\n"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
```

```
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous ListTablesà la section Référence des AWS CLI commandes.

`PutItem`

L'exemple de code suivant montre comment utiliser`PutItem`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -i item  -- Path to json file containing the item values.
#
```

```
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_put_item() {
  local table_name item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -i item  -- Path to json file containing the item values."
    echo ""
  }

  while getopts "n:i:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
```

```
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    item:   $item"
  iecho ""
  iecho ""

  response=$(aws dynamodb put-item \
    --table-name "$table_name" \
    --item file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.$response"
    return 1
  fi

  return 0

}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
```

```
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous PutItemà la section Référence des AWS CLI commandes.

`Query`

L'exemple de code suivant montre comment utiliser`Query`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k key_condition_expression -- The key condition expression.
#       -a attribute_names -- Path to JSON file containing the attribute names.
#       -v attribute_values -- Path to JSON file containing the attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function dynamodb_query() {
  local table_name key_condition_expression attribute_names attribute_values
 projection_expression response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
```

```
 # Function usage explanation
 ####################################
 function usage() {
   echo "function dynamodb_query"
   echo "Query a DynamoDB table."
   echo " -n table_name  -- The name of the table."
   echo " -k key_condition_expression -- The key condition expression."
   echo " -a attribute_names -- Path to JSON file containing the attribute names."
   echo " -v attribute_values -- Path to JSON file containing the attribute
values."
   echo " [-p projection_expression]  -- Optional projection expression."
   echo ""
 }

 while getopts "n:k:a:v:p:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     k) key_condition_expression="${OPTARG}" ;;
     a) attribute_names="${OPTARG}" ;;
     v) attribute_values="${OPTARG}" ;;
     p) projection_expression="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$key_condition_expression" ]]; then
   errecho "ERROR: You must provide a key condition expression with the -k
parameter."
   usage
```

```
    return 1
  fi

  if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
  fi

  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values")
  else
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi
```

```
  return 0
}
```

- Pour plus d'informations sur l'API, consultez Query dans la Référence des commandes AWS CLI .

`Scan`

L'exemple de code suivant montre comment utiliser`Scan`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_scan
#
# This function scans a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -f filter_expression  -- The filter expression.
#       -a expression_attribute_names -- Path to JSON file containing the expression
 attribute names.
#       -v expression_attribute_values -- Path to JSON file containing the
 expression attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_scan() {
```

```
 local table_name filter_expression expression_attribute_names
expression_attribute_values projection_expression response
 local option OPTARG # Required to use getopts command in a function.

 # ######################################
 # Function usage explanation
 ######################################
 function usage() {
   echo "function dynamodb_scan"
   echo "Scan a DynamoDB table."
   echo " -n table_name  -- The name of the table."
   echo " -f filter_expression  -- The filter expression."
   echo " -a expression_attribute_names -- Path to JSON file containing the
expression attribute names."
   echo " -v expression_attribute_values -- Path to JSON file containing the
expression attribute values."
   echo " [-p projection_expression]  -- Optional projection expression."
   echo ""
 }

 while getopts "n:f:a:v:p:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     f) filter_expression="${OPTARG}" ;;
     a) expression_attribute_names="${OPTARG}" ;;
     v) expression_attribute_values="${OPTARG}" ;;
     p) projection_expression="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
```

```
 fi

 if [[ -z "$filter_expression" ]]; then
   errecho "ERROR: You must provide a filter expression with the -f parameter."
   usage
   return 1
 fi

 if [[ -z "$expression_attribute_names" ]]; then
   errecho "ERROR: You must provide expression attribute names with the -a
parameter."
   usage
   return 1
 fi

 if [[ -z "$expression_attribute_values" ]]; then
   errecho "ERROR: You must provide expression attribute values with the -v
parameter."
   usage
   return 1
 fi

 if [[ -z "$projection_expression" ]]; then
   response=$(aws dynamodb scan \
     --table-name "$table_name" \
     --filter-expression "$filter_expression" \
     --expression-attribute-names file://"$expression_attribute_names" \
     --expression-attribute-values file://"$expression_attribute_values")
 else
   response=$(aws dynamodb scan \
     --table-name "$table_name" \
     --filter-expression "$filter_expression" \
     --expression-attribute-names file://"$expression_attribute_names" \
     --expression-attribute-values file://"$expression_attribute_values" \
     --projection-expression "$projection_expression")
 fi

 local error_code=${?}

 if [[ $error_code -ne 0 ]]; then
   aws_cli_error_log $error_code
   errecho "ERROR: AWS reports scan operation failed.$response"
   return 1
 fi
```

```
  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
```

```
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus d'informations sur l'API, consultez Scan dans la Référence des commandes AWS CLI .

`UpdateItem`

L'exemple de code suivant montre comment utiliser`UpdateItem`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 update.
#       -e update expression  -- An expression that defines one or more attributes
 to be updated.
#       -v values  -- Path to json file containing the update values.
```

```
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_update_item() {
  local table_name keys update_expression values response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_update_item"
    echo "Update an item in a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to update."
    echo " -e update expression  -- An expression that defines one or more
 attributes to be updated."
    echo " -v values  -- Path to json file containing the update values."
    echo ""
  }

  while getopts "n:k:e:v:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      e) update_expression="${OPTARG}" ;;
      v) values="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
```

```
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$keys" ]]; then
  errecho "ERROR: You must provide a keys json file path the -k parameter."
  usage
  return 1
fi
if [[ -z "$update_expression" ]]; then
  errecho "ERROR: You must provide an update expression with the -e parameter."
  usage
  return 1
fi

if [[ -z "$values" ]]; then
  errecho "ERROR: You must provide a values json file path the -v parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    table_name:   $table_name"
iecho "    keys:   $keys"
iecho "    update_expression:   $update_expression"
iecho "    values:   $values"

response=$(aws dynamodb update-item \
  --table-name "$table_name" \
  --key file://"$keys" \
  --update-expression "$update_expression" \
  --expression-attribute-values file://"$values")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports update-item operation failed.$response"
  return 1
fi

return 0
```

```
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
```

```
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateItemà la section Référence des AWS CLI commandes.

Scénarios

Scénarios avancés relatifs à l'indice secondaire mondial

L'exemple de code suivant montre comment utiliser des configurations avancées d'index secondaire global.

- Créez une table avec plusieurs GSIs.
- Créez une table avec une capacité à la demande et un GSI.
- Placez les éléments dans un tableau comportant plusieurs éléments GSIs.
- Interrogez plusieurs GSIs avec des conditions différentes.

AWS CLI avec le script Bash

Créez une table avec plusieurs GSIs.

```
# Create a table with multiple GSIs
aws dynamodb create-table \
    --table-name MusicLibrary \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=AlbumTitle,AttributeType=S \
        AttributeName=Genre,AttributeType=S \
        AttributeName=Year,AttributeType=N \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"AlbumIndex\",
                \"KeySchema\": [{\"AttributeName\":\"AlbumTitle\",\"KeyType\":\"HASH
\"}],
                \"Projection\": {\"ProjectionType\":\"ALL\"}
            },
            {
                \"IndexName\": \"GenreYearIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"Year\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"INCLUDE\",\"NonKeyAttributes
\":[\"Artist\",\"SongTitle\"]}
            }
        ]"
```

Créez une table avec une capacité à la demande et un GSI.

```
# Create a table with on-demand capacity and GSI
aws dynamodb create-table \
    --table-name MusicOnDemand \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=Genre,AttributeType=S \
    --key-schema \
```

```
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"GenreIndex\",
                \"KeySchema\": [{\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"}],
                \"Projection\": {\"ProjectionType\":\"ALL\"}
            }
        ]"
```

Placez les éléments dans un tableau comportant plusieurs éléments GSIs.

```
# Add items to MusicLibrary table
aws dynamodb put-item \
    --table-name MusicLibrary \
    --item '{
        "Artist": {"S": "The Beatles"},
        "SongTitle": {"S": "Hey Jude"},
        "AlbumTitle": {"S": "Past Masters"},
        "Genre": {"S": "Rock"},
        "Year": {"N": "1968"}
    }'

aws dynamodb put-item \
    --table-name MusicLibrary \
    --item '{
        "Artist": {"S": "Miles Davis"},
        "SongTitle": {"S": "So What"},
        "AlbumTitle": {"S": "Kind of Blue"},
        "Genre": {"S": "Jazz"},
        "Year": {"N": "1959"}
    }'
```

Interrogez les éléments d'une table contenant plusieurs éléments GSIs.

```
# Query the AlbumIndex GSI
echo "Querying AlbumIndex GSI:"
aws dynamodb query \
    --table-name MusicLibrary \
```

```
    --index-name AlbumIndex \
    --key-condition-expression "AlbumTitle = :album" \
    --expression-attribute-values '{":album":{"S":"Kind of Blue"}}'

# Query the GenreYearIndex GSI with a range condition
echo "Querying GenreYearIndex GSI with range condition:"
aws dynamodb query \
    --table-name MusicLibrary \
    --index-name GenreYearIndex \
    --key-condition-expression "Genre = :genre AND #yr > :year" \
    --expression-attribute-names '{"#yr": "Year"}' \
    --expression-attribute-values '{":genre":{"S":"Rock"},":year":{"N":"1965"}}'
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateTable
  - PutItem
  - Interrogation

Création et gestion des tables globales MRSC

L'exemple de code suivant montre comment créer et gérer des tables globales DynamoDB avec Multi-Region Strong Consistency (MRSC).

- Créez un tableau avec une forte cohérence multirégionale.
- Vérifiez la configuration MRSC et l'état de la réplique.
- Testez une forte cohérence entre les régions grâce à des lectures immédiates.
- Effectuez des écritures conditionnelles avec les garanties MRSC.
- Nettoyez les ressources des tables mondiales du MRSC.

AWS CLI avec le script Bash

Créez un tableau avec une forte cohérence multirégionale.

```
# Step 1: Create a new table in us-east-2 (primary region for MRSC)
# Note: Table must be empty when enabling MRSC
aws dynamodb create-table \
```

```
    --table-name MusicTable \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --region us-east-2

# Wait for table to become active
aws dynamodb wait table-exists --table-name MusicTable --region us-east-2

# Step 2: Add replica and witness with Multi-Region Strong Consistency
# MRSC requires exactly three replicas in supported regions
aws dynamodb update-table \
    --table-name MusicTable \
    --replica-updates '[{"Create": {"RegionName": "us-east-1"}}]' \
    --global-table-witness-updates '[{"Create": {"RegionName": "us-west-2"}}]' \
    --multi-region-consistency STRONG \
    --region us-east-2
```

Vérifiez la configuration MRSC et l'état de la réplique.

```
# Verify the global table configuration and MRSC setting
aws dynamodb describe-table \
    --table-name MusicTable \
    --region us-east-2 \
    --query 'Table.
{TableName:TableName,TableStatus:TableStatus,MultiRegionConsistency:MultiRegionConsistency,R
{Region:RegionName,Status:ReplicaStatus}}'
```

Testez une forte cohérence avec des lectures immédiates dans toutes les régions.

```
# Write an item to the primary region
aws dynamodb put-item \
    --table-name MusicTable \
    --item '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"},"Album":
 {"S":"The Beatles 1967-1970"},"Year": {"N":"1968"}}' \
    --region us-east-2
```

```
# Read the item from replica region to verify strong consistency (cannot read or
 write to witness)
# No wait time needed - MRSC provides immediate consistency
echo "Reading from us-east-1 (immediate consistency):"
aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"}}' \
    --consistent-read \
    --region us-east-1
```

Effectuez des écritures conditionnelles avec les garanties MRSC.

```
# Perform a conditional update from a different region
# This demonstrates that conditions work consistently across all regions
aws dynamodb update-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"}}' \
    --update-expression "SET #rating = :rating" \
    --condition-expression "attribute_exists(Artist)" \
    --expression-attribute-names '{"#rating": "Rating"}' \
    --expression-attribute-values '{":rating": {"N":"5"}}' \
    --region us-east-1
```

Nettoyez les ressources des tables mondiales du MRSC.

```
# Remove replica tables (must be done before deleting the primary table)
aws dynamodb update-table \
    --table-name MusicTable \
    --replica-updates '[{"Delete": {"RegionName": "us-east-1"}}]' \
    --global-table-witness-updates '[{"Delete": {"RegionName": "us-west-2"}}]' \
    --region us-east-2

# Wait for replicas to be deleted
echo "Waiting for replicas to be deleted..."
sleep 30

# Delete the primary table
aws dynamodb delete-table \
    --table-name MusicTable \
    --region us-east-2
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateTable
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - UpdateItem
  - UpdateTable

Gérer les index secondaires globaux

L'exemple de code suivant montre comment gérer le cycle de vie complet des index secondaires globaux.

- Créez une table avec un index secondaire global.
- Ajoutez un nouveau GSI à une table existante.
- Mettre à jour (augmenter) le débit à chaud du GSI.
- Interrogez les données en utilisant GSIs.
- Supprimez un GSI.

AWS CLI avec le script Bash

Créez une table avec un index secondaire global.

```
# Create a table with a GSI
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=AlbumTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
```

```
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "IndexName=AlbumIndex,\
        KeySchema=[{AttributeName=AlbumTitle,KeyType=HASH}],\
        Projection={ProjectionType=ALL}"
```

Ajoutez un nouveau GSI (à la demande) à une table existante.

```
# Add a new GSI to an existing table
aws dynamodb update-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Genre,AttributeType=S \
    --global-secondary-index-updates \
        "[{\"Create\":{\"IndexName\":\"GenreIndex\",\
        \"KeySchema\":[{\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"}],\
        \"Projection\":{\"ProjectionType\":\"ALL\"}}}]"
```

Mettre à jour (augmenter) le débit à chaud du GSI.

```
# Increase the warm throughput of a GSI (default values are 12k reads, 4k writes)
aws dynamodb update-table \
    --table-name MusicCollection \
    --global-secondary-index-updates \
        "[{\"Update\":{\"IndexName\":\"AlbumIndex\",\
        \"WarmThroughput\":{\"ReadUnitsPerSecond\":15000,\"WriteUnitsPerSecond
\":6000}}}]"
```

Interrogez les données en utilisant GSIs.

```
# Query the AlbumIndex GSI
aws dynamodb query \
    --table-name MusicCollection \
    --index-name AlbumIndex \
    --key-condition-expression "AlbumTitle = :album" \
    --expression-attribute-values '{":album":{"S":"Let It Be"}}'

# Query the GenreIndex GSI
aws dynamodb query \
```

```
    --table-name MusicCollection \
    --index-name GenreIndex \
    --key-condition-expression "Genre = :genre" \
    --expression-attribute-values '{":genre":{"S":"Jazz"}}'
```

Supprimez un GSI.

```
# Delete a GSI from a table
aws dynamodb update-table \
    --table-name MusicCollection \
    --global-secondary-index-updates \
        "[{\"Delete\":{\"IndexName\":\"GenreIndex\"}}]"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateTable
  - DeleteTable
  - Interrogation
  - UpdateTable

Gérer les politiques basées sur les ressources

L'exemple de code suivant montre comment gérer le cycle de vie complet des politiques basées sur les ressources pour les tables DynamoDB.

- Créez une table avec une politique de ressources.
- Obtenez une politique en matière de ressources.
- Mettez à jour une politique de ressources.
- Supprimez une politique de ressources.

AWS CLI avec le script Bash

Créez une table avec une politique de ressources.

```
# Step 1: Create a DynamoDB table
aws dynamodb create-table \
```

```
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST

# Step 2: Create a resource-based policy document
cat > policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/DynamoDBReadOnly"
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection"
    }
  ]
}
EOF

# Step 3: Attach the resource-based policy to the table
aws dynamodb put-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --policy file://policy.json
```

Obtenez une politique en matière de ressources.

```
# Get the resource-based policy attached to a table
aws dynamodb get-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

Mettez à jour une politique de ressources.

```
# Step 1: Create an updated policy document
cat > updated-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/DynamoDBReadOnly",
          "arn:aws:iam::123456789012:role/DynamoDBAnalytics"
        ]
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection"
    }
  ]
}
EOF

# Step 2: Update the resource-based policy on the table
aws dynamodb put-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --policy file://updated-policy.json
```

Supprimez une politique de ressources.

```
# Delete the resource-based policy from a table
aws dynamodb delete-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

- Pour plus d’informations sur l’API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .

- CreateTable
- DeleteResourcePolicy
- GetResourcePolicy
- PutResourcePolicy

Configurer le contrôle d'accès basé sur les attributs

L'exemple de code suivant montre comment implémenter le contrôle d'accès basé sur les attributs (ABAC) pour DynamoDB.

- Créez une politique IAM pour ABAC.
- Créez des tableaux avec des balises pour différents départements.
- Répertoriez et filtrez les tableaux en fonction des balises.

AWS CLI avec le script Bash

Créez une politique IAM pour ABAC.

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

Créez des tableaux avec des balises pour différents départements.

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
```

```
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
        Key=Environment,Value=Production
```

Répertoriez et filtrez les tableaux en fonction des balises.

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .

- CreatePolicy
- CreateTable
- ListTables

Travailler avec des index secondaires locaux

L'exemple de code suivant montre comment créer et interroger des tables avec des index secondaires locaux.

- Créez une table avec un index secondaire local (LSI).
- Créez un tableau à plusieurs LSIs avec différents types de projection.
- Interrogez les données en utilisant LSIs.

AWS CLI avec le script Bash

Créez une table avec un index secondaire local.

```
# Create a table with a Local Secondary Index
aws dynamodb create-table \
    --table-name CustomerOrders \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
        AttributeName=OrderID,AttributeType=S \
        AttributeName=OrderDate,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
        AttributeName=OrderID,KeyType=RANGE \
    --local-secondary-indexes \
        "IndexName=OrderDateIndex,\
        KeySchema=[{AttributeName=CustomerID,KeyType=HASH},
{AttributeName=OrderDate,KeyType=RANGE}],\
        Projection={ProjectionType=ALL}" \
    --billing-mode PAY_PER_REQUEST
```

Créez une table avec plusieurs LSIs.

```
# Create a table with multiple Local Secondary Indexes
aws dynamodb create-table \
```

```
    --table-name CustomerDetails \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
        AttributeName=Name,AttributeType=S \
        AttributeName=Email,AttributeType=S \
        AttributeName=RegistrationDate,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
        AttributeName=Name,KeyType=RANGE \
    --local-secondary-indexes \
        "[
            {
                \"IndexName\": \"EmailIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"CustomerID\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"Email\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"INCLUDE\",\"NonKeyAttributes
\":[\"Address\",\"Phone\"]}
            },
            {
                \"IndexName\": \"RegistrationIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"CustomerID\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"RegistrationDate\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"KEYS_ONLY\"}
            }
        ]" \
    --billing-mode PAY_PER_REQUEST
```

Interrogez les données en utilisant LSIs.

```
# Query the OrderDateIndex LSI
aws dynamodb query \
    --table-name CustomerOrders \
    --index-name OrderDateIndex \
    --key-condition-expression "CustomerID = :custId AND OrderDate BETWEEN :date1
 AND :date2" \
    --expression-attribute-values '{
        ":custId": {"S": "C1"},
        ":date1": {"S": "2023-01-01"},
```

```
        ":date2": {"S": "2023-02-01"}
    }'

# Query with a filter expression
aws dynamodb query \
    --table-name CustomerOrders \
    --index-name OrderDateIndex \
    --key-condition-expression "CustomerID = :custId" \
    --filter-expression "Amount > :amount" \
    --expression-attribute-values '{
        ":custId": {"S": "C1"},
        ":amount": {"N": "150"}
    }'
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateTable
  - Interrogation

Travaillez avec Streams et Time-to-Live

L'exemple de code suivant montre comment gérer les flux et les fonctionnalités de DynamoDB. Time-to-Live

- Créez une table avec Streams activé.
- Décrivez les flux.
- Créez une fonction Lambda pour le traitement des flux.
- Activez le TTL sur une table.
- Ajoutez des éléments dotés d'attributs TTL.
- Décrivez les paramètres TTL.

AWS CLI avec le script Bash

Créez une table avec Streams activé.

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
```

```
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

Décrivez les flux.

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"

# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

Créez une fonction Lambda pour Streams.

```
# Step 1: Create an IAM role for the Lambda function
cat > trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

Activez le TTL sur une table.

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST

# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo

# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

Ajoutez des éléments dotés d'attributs TTL.

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

Décrivez les paramètres TTL.

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AttachRolePolicy
  - CreateRole
  - CreateTable
  - DescribeTable
  - DescribeTimeToLive
  - PutItem

- UpdateTimeToLive

Travaillez avec des tables globales et la cohérence éventuelle de la réplication multirégionale (MREC)

L'exemple de code suivant montre comment gérer les tables globales DynamoDB avec réplication multirégionale avec cohérence éventuelle (MREC).

- Créez une table avec la réplication multirégionale (MREC).
- Placez et récupérez des objets sur des répliques de tables.
- Supprimez les répliques one-by-one.
- Nettoyez en supprimant le tableau.

AWS CLI avec le script Bash

Créez une table avec réplication multirégionale.

```
# Step 1: Create a new table (MusicTable) in US East (Ohio), with DynamoDB Streams
 enabled (NEW_AND_OLD_IMAGES)
aws dynamodb create-table \
    --table-name MusicTable \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES \
    --region us-east-2

# Step 2: Create an identical MusicTable table in US East (N. Virginia)
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Create": {
        "RegionName": "us-east-1"
      }
```

```
    }
  ]
}' \
--region us-east-2

# Step 3: Create a table in Europe (Ireland)
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Create": {
        "RegionName": "eu-west-1"
      }
    }
  ]
}' \
--region us-east-2
```

Décrivez le tableau multirégional.

```
# Step 4: View the list of replicas created using describe-table
aws dynamodb describe-table \
    --table-name MusicTable \
    --region us-east-2 \
    --query 'Table.
{TableName:TableName,TableStatus:TableStatus,MultiRegionConsistency:MultiRegionConsistency,R
{Region:RegionName,Status:ReplicaStatus}}'
```

Placez les objets dans une réplique de table.

```
# Step 5: To verify that replication is working, add a new item to the Music table
 in US East (Ohio)
aws dynamodb put-item \
    --table-name MusicTable \
    --item '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region us-east-2
```

Récupérez des objets à partir de répliques de tables.

```
# Step 6: Wait for a few seconds, and then check to see whether the item has been
# successfully replicated to US East (N. Virginia) and Europe (Ireland)
aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region us-east-1

aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region eu-west-1
```

Supprimez les répliques.

```
# Step 7: Delete the replica table in Europe (Ireland) Region
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Delete": {
        "RegionName": "eu-west-1"
      }
    }
  ]
}' \
--region us-east-2

# Delete the replica table in US East (N. Virginia) Region
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Delete": {
        "RegionName": "us-east-1"
      }
    }
  ]
}' \
--region us-east-2
```

Nettoyez en supprimant le tableau.

```
# Clean up: Delete the primary table
aws dynamodb delete-table --table-name MusicTable --region us-east-2

echo "Global table demonstration complete."
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateTable
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - UpdateTable

Utiliser le balisage des ressources

L'exemple de code suivant montre comment gérer les balises pour les ressources DynamoDB.

- Créez un tableau avec des balises.
- Répertorie les balises d'une ressource.
- Ajoutez des balises à une ressource.
- Supprimer des balises d'une ressource.
- Filtrez les tableaux par balises.

AWS CLI avec le script Bash

Créez un tableau avec des balises.

```
# Create a table with tags
aws dynamodb create-table \
    --table-name TaggedTable \
```

```
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Environment,Value=Production \
        Key=Project,Value=Analytics \
        Key=Owner,Value=DataTeam
```

Répertorie les balises d'une ressource.

```
# Get the table ARN
TABLE_ARN=$(aws dynamodb describe-table \
    --table-name TaggedTable \
    --query "Table.TableArn" \
    --output text)

# List tags for the table
aws dynamodb list-tags-of-resource \
    --resource-arn $TABLE_ARN
```

Ajoutez des balises à une ressource.

```
# Add tags to an existing table
aws dynamodb tag-resource \
    --resource-arn $TABLE_ARN \
    --tags \
        Key=CostCenter,Value=12345 \
        Key=BackupSchedule,Value=Daily
```

Supprimer des balises d'une ressource.

```
# Remove tags from a table
aws dynamodb untag-resource \
    --resource-arn $TABLE_ARN \
    --tag-keys Owner BackupSchedule
```

Filtrez les tableaux par balises.

```
# Create another table with different tags
aws dynamodb create-table \
    --table-name AnotherTaggedTable \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Environment,Value=Development \
        Key=Project,Value=Testing

# Wait for table to become active
aws dynamodb wait table-exists --table-name AnotherTaggedTable

# List all tables
echo "All tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nFiltering tables by Environment=Production tag:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# Find tables with specific tag
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateTable
  - ListTagsOfResource

- TagResource
- UntagResource

Travaillez avec le chiffrement des tables

L'exemple de code suivant montre comment gérer les options de chiffrement pour les tables DynamoDB.

- Créez une table avec le chiffrement par défaut.
- Créez une table avec une clé CMK gérée par le client.
- Mettez à jour les paramètres de chiffrement des tables.
- Décrivez le chiffrement des tables.

AWS CLI avec le script Bash

Créez une table avec le chiffrement par défaut.

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

Créez une table avec une clé CMK gérée par le client.

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)
```

```
# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Mettez à jour le chiffrement des tables.

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Décrivez le chiffrement des tables.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

EC2 Exemples d'utilisation d'Amazon AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash avec Amazon EC2.

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Principes de base
- Actions
- Scénarios

Principes de base

Principes de base

L'exemple de code suivant illustre comment :

- Créez une paire de clés et un groupe de sécurité.
- Sélectionnez une Amazon Machine Image (AMI) et un type d'instance compatible, puis créez une instance.
- Arrêtez l'instance, puis redémarrez-la.
- Associez une adresse IP Elastic à votre instance
- Connectez-vous à votre instance avec SSH, puis nettoyez les ressources.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exécutez un scénario interactif à une invite de commande.

```
###############################################################################
# function get_started_with_ec2_instances
#
# Runs an interactive scenario that shows how to get started using EC2 instances.
#
#     "EC2 access" permissions are needed to run this code.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function get_started_with_ec2_instances() {
  # Requires version 4 for mapfile.
  local required_version=4.0

  # Get the current Bash version
  # Check if BASH_VERSION is set
  local current_version
  if [[ -n "$BASH_VERSION" ]]; then
    # Convert BASH_VERSION to a number for comparison
    current_version=$BASH_VERSION
  else
    # Get the current Bash version using the bash command
    current_version=$(bash --version | head -n 1 | awk '{ print $4 }')
  fi

  # Convert version strings to numbers for comparison
  local required_version_num current_version_num
  required_version_num=$(echo "$required_version" | awk -F. '{ print ($1 * 10000) +
($2 * 100) + $3 }')
  current_version_num=$(echo "$current_version" | awk -F. '{ print ($1 * 10000) +
($2 * 100) + $3 }')
```

```
  # Compare versions
  if ((current_version_num < required_version_num)); then
    echo "Error: This script requires Bash version $required_version or higher."
    echo "Your current Bash version is number is $current_version."
    exit 1
  fi

  {
    if [ "$EC2_OPERATIONS_SOURCED" != "True" ]; then

      source ./ec2_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started with
instances demo."
  echo_repeat "*" 88
  echo

  echo "Let's create an RSA key pair that you can be use to securely connect to "
  echo "your EC2 instance."

  echo -n "Enter a unique name for your key: "
  get_input
  local key_name
  key_name=$get_input_result

  local temp_dir
  temp_dir=$(mktemp -d)
  local key_file_name="$temp_dir/${key_name}.pem"

  if ec2_create_keypair -n "${key_name}" -f "${key_file_name}"; then
    echo "Created a key pair $key_name and saved the private key to $key_file_name"
    echo
  else
    errecho "The key pair failed to create. This demo will exit."
    return 1
  fi

  chmod 400 "${key_file_name}"

  if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then
    local keys_and_fingerprints
```

```
    keys_and_fingerprints="$(ec2_describe_key_pairs)" && {
      local image_name_and_id
      while IFS=$'\n' read -r image_name_and_id; do
        local entries
        IFS=$'\t' read -ra entries <<<"$image_name_and_id"
        echo "Found rsa key ${entries[0]} with fingerprint:"
        echo "     ${entries[1]}"
      done <<<"$keys_and_fingerprints"

    }
  fi

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's create a security group to manage access to your instance."
  echo -n "Enter a unique name for your security group: "
  get_input
  local security_group_name
  security_group_name=$get_input_result
  local security_group_id
  security_group_id=$(ec2_create_security_group -n "$security_group_name" \
    -d "Security group for EC2 instance") || {
    errecho "The security failed to create. This demo will exit."
    clean_up "$key_name" "$key_file_name"
    return 1
  }

  echo "Security group created with ID $security_group_id"
  echo

  local public_ip
  public_ip=$(curl -s http://checkip.amazonaws.com)

  echo "Let's add a rule to allow SSH only from your current IP address."
  echo "Your public IP address is $public_ip"
  echo -n "press return to add this rule to your security group."
  get_input

  if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i "$public_ip"
-p tcp -f 22 -t 22; then
    errecho "The security group rules failed to update. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
```

```
  fi

  echo "Security group rules updated"

  local security_group_description
  security_group_description="$(ec2_describe_security_groups -g
 "${security_group_id}")" || {
    errecho "Failed to describe security groups. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }

  mapfile -t parameters <<<"$security_group_description"
  IFS=$'\t' read -ra entries <<<"${parameters[0]}"
  echo "Security group: ${entries[0]}"
  echo "    ID: ${entries[1]}"
  echo "    VPC: ${entries[2]}"
  echo "Inbound permissions:"
  IFS=$'\t' read -ra entries <<<"${parameters[1]}"
  echo "    IpProtocol: ${entries[0]}"
  echo "    FromPort: ${entries[1]}"
  echo "    ToPort: ${entries[2]}"
  echo "    CidrIp: ${parameters[2]}"

  local parameters
  parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
    errecho "Failed to get parameters. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1

  }

  local image_ids=""
  mapfile -t parameters <<<"$parameters"
  for image_name_and_id in "${parameters[@]}"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
      image_ids+="${values[1]} "
    fi
  done

  local images
  images="$(ec2_describe_images -i "$image_ids")" || {
```

```
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1

  }

  new_line_and_tab_to_list "$images"
  local images=("${list_result[@]}")

  # Get the size of the array
  local images_count=${#images[@]}

  if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  fi

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some options:"
  for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
  done

  integer_input "Please enter the number of the AMI you want to use: " 1
"$((images_count / 3))"
  local choice=$get_input_result
  choice=$(((choice - 1) * 3))

  echo "Great choice."
  echo

  local architecture=${images[$((choice + 1))]}
  local image_id=${images[$((choice + 2))]}
  echo "Here are some instance types that support the ${architecture} architecture
of the image:"
  response="$(ec2_describe_instance_types -a "${architecture}" -t
"*.micro,*.small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }
```

```
  local instance_types
  mapfile -t instance_types <<<"$response"

  # Get the size of the array
  local instance_types_count=${#instance_types[@]}

  echo "Here are some options:"
  for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
  done

  integer_input "Which one do you want to use? " 1 "${#instance_types[@]}
"
  choice=$get_input_result
  local instance_type=${instance_types[$((choice - 1))]}
  echo "Another great choice."
  echo

  echo "Creating your instance and waiting for it to start..."
  local instance_id
  instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k "$key_name"
 -s "$security_group_id") || {
    errecho "Failed to run instance. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }

  ec2_wait_for_instance_running -i "$instance_id"
  echo "Your instance is ready:"
  echo

  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  echo
  print_instance_details "${instance_details}"

  local public_ip
  public_ip=$(echo "${instance_details}" | awk '{print $6}')
  echo
  echo "You can use SSH to connect to your instance"
  echo "If the connection attempt times out, you might have to manually update the
 SSH ingress rule"
```

```
  echo "for your IP address in the AWS Management Console."
  connect_to_instance "$key_file_name" "$public_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's stop and start your instance to see what changes."
  echo "Stopping your instance and waiting until it's stopped..."
  ec2_stop_instances -i "$instance_id"
  ec2_wait_for_instance_stopped -i "$instance_id"

  echo "Your instance is stopped. Restarting..."

  ec2_start_instances -i "$instance_id"
  ec2_wait_for_instance_running -i "$instance_id"

  echo "Your instance is running again."
  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  print_instance_details "${instance_details}"

  public_ip=$(echo "${instance_details}" | awk '{print $6}')

  echo "Every time your instance is restarted, its public IP address changes"
  connect_to_instance "$key_file_name" "$public_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "You can allocate an Elastic IP address and associate it with your instance"
  echo "to keep a consistent IP address even when your instance restarts."

  local result
  result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
```

```
  }

  local elastic_ip allocation_id
  elastic_ip=$(echo "$result" | awk '{print $1}')
  allocation_id=$(echo "$result" | awk '{print $2}')

  echo "Allocated static Elastic IP address: $elastic_ip"

  local association_id
  association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id") || {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
"$allocation_id"
    return 1
  }

  echo "Associated your Elastic IP with your instance."
  echo "You can now use SSH to connect to your instance by using the Elastic IP."
  connect_to_instance "$key_file_name" "$elastic_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's stop and start your instance to see what changes."
  echo "Stopping your instance and waiting until it's stopped..."
  ec2_stop_instances -i "$instance_id"
  ec2_wait_for_instance_stopped -i "$instance_id"

  echo "Your instance is stopped. Restarting..."

  ec2_start_instances -i "$instance_id"
  ec2_wait_for_instance_running -i "$instance_id"

  echo "Your instance is running again."
  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  print_instance_details "${instance_details}"

  echo "Because you have associated an Elastic IP with your instance, you can"
  echo "connect by using a consistent IP address after the instance restarts."
```

```
  connect_to_instance "$key_file_name" "$elastic_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  if yes_no_input "Do you want to delete the resources created in this demo: (y/n)
 "; then
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \
      "$allocation_id" "$association_id"
  else
    echo "The following resources were not deleted."
    echo "Key pair: $key_name"
    echo "Key file: $key_file_name"
    echo "Security group: $security_group_id"
    echo "Instance: $instance_id"
    echo "Elastic IP address: $elastic_ip"
  fi
}

###############################################################################
# function clean_up
#
# This function cleans up the created resources.
#     $1 - The name of the ec2 key pair to delete.
#     $2 - The name of the key file to delete.
#     $3 - The ID of the security group to delete.
#     $4 - The ID of the instance to terminate.
#     $5 - The ID of the elastic IP address to release.
#     $6 - The ID of the elastic IP address to disassociate.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function clean_up() {
  local result=0
  local key_pair_name=$1
  local key_file_name=$2
  local security_group_id=$3
  local instance_id=$4
  local allocation_id=$5
```

```
  local association_id=$6

  if [ -n "$association_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_disassociate_address -a "$association_id"); then
      echo "Disassociated elastic IP address with ID $association_id"
    else
      errecho "The elastic IP address disassociation failed."
      result=1
    fi
  fi

  if [ -n "$allocation_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_release_address -a "$allocation_id"); then
      echo "Released elastic IP address with ID $allocation_id"
    else
      errecho "The elastic IP address release failed."
      result=1
    fi
  fi

  if [ -n "$instance_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_terminate_instances -i "$instance_id"); then
      echo "Started terminating instance with ID $instance_id"

      ec2_wait_for_instance_terminated -i "$instance_id"
    else
      errecho "The instance terminate failed."
      result=1
    fi
  fi

  if [ -n "$security_group_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_security_group -i "$security_group_id"); then
      echo "Deleted security group with ID $security_group_id"
    else
      errecho "The security group delete failed."
      result=1
    fi
  fi
```

```
  if [ -n "$key_pair_name" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_keypair -n "$key_pair_name"); then
      echo "Deleted key pair named $key_pair_name"
    else
      errecho "The key pair delete failed."
      result=1
    fi
  fi

  if [ -n "$key_file_name" ]; then
    rm -f "$key_file_name"
  fi

  return $result
}

###############################################################################
# function ssm_get_parameters_by_path
#
# This function retrieves one or more parameters from the AWS Systems Manager
 Parameter Store
# by specifying a parameter path.
#
# Parameters:
#       -p parameter_path - The path of the parameter(s) to retrieve.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ssm_get_parameters_by_path() {
  local parameter_path response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ssm_get_parameters_by_path"
    echo "Retrieves one or more parameters from the AWS Systems Manager Parameter
 Store by specifying a parameter path."
    echo "  -p parameter_path - The path of the parameter(s) to retrieve."
    echo ""
  }
```

```
  # Retrieve the calling parameters.
  while getopts "p:h" option; do
    case "${option}" in
      p) parameter_path="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$parameter_path" ]]; then
    errecho "ERROR: You must provide a parameter path with the -p parameter."
    usage
    return 1
  fi

  response=$(aws ssm get-parameters-by-path \
    --path "$parameter_path" \
    --query "Parameters[*].[Name, Value]" \
    --output text) || {
    aws_cli_error_log $?
    errecho "ERROR: AWS reports get-parameters-by-path operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

###############################################################################
# function print_instance_details
#
# This function prints the details of an Amazon Elastic Compute Cloud (Amazon EC2)
 instance.
#
# Parameters:
```

```
#       instance_details - The instance details in the format "InstanceId ImageId
 InstanceType KeyName VpcId PublicIpAddress State.Name".
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function print_instance_details() {
  local instance_details="$1"

  if [[ -z "${instance_details}" ]]; then
    echo "Error: Missing required instance details argument."
    return 1
  fi

  local instance_id image_id instance_type key_name vpc_id public_ip state
  instance_id=$(echo "${instance_details}" | awk '{print $1}')
  image_id=$(echo "${instance_details}" | awk '{print $2}')
  instance_type=$(echo "${instance_details}" | awk '{print $3}')
  key_name=$(echo "${instance_details}" | awk '{print $4}')
  vpc_id=$(echo "${instance_details}" | awk '{print $5}')
  public_ip=$(echo "${instance_details}" | awk '{print $6}')
  state=$(echo "${instance_details}" | awk '{print $7}')

  echo "    ID: ${instance_id}"
  echo "    Image ID: ${image_id}"
  echo "    Instance type: ${instance_type}"
  echo "    Key name: ${key_name}"
  echo "    VPC ID: ${vpc_id}"
  echo "    Public IP: ${public_ip}"
  echo "    State: ${state}"

  return 0
}

###############################################################################
# function connect_to_instance
#
# This function displays the public IP address of an Amazon Elastic Compute Cloud
 (Amazon EC2) instance and prompts the user to connect to the instance via SSH.
#
# Parameters:
#       $1 - The name of the key file used to connect to the instance.
#       $2 - The public IP address of the instance.
```

```
#
# Returns:
#       None
###############################################################################
function connect_to_instance() {
  local key_file_name="$1"
  local public_ip="$2"

  # Validate the input parameters
  if [[ -z "$key_file_name" ]]; then
    echo "ERROR: You must provide a key file name as the first argument." >&2
    return 1
  fi

  if [[ -z "$public_ip" ]]; then
    echo "ERROR: You must provide a public IP address as the second argument." >&2
    return 1
  fi

  # Display the public IP address and connection command
  echo "To connect, run the following command:"
  echo "    ssh -i ${key_file_name} ec2-user@${public_ip}"

  # Prompt the user to connect to the instance
  if yes_no_input "Do you want to connect now? (y/n) "; then
    echo "After you have connected, you can return to this example by typing 'exit'"
    ssh -i "${key_file_name}" ec2-user@"${public_ip}"
  fi
}

###############################################################################
# function get_input
#
# This function gets user input from the command line.
#
# Outputs:
#   User input to stdout.
#
# Returns:
#       0
###############################################################################
function get_input() {

  if [ -z "${mock_input+x}" ]; then
```

```
    read -r get_input_result
  else

    if [ "$mock_input_array_index" -lt ${#mock_input_array[@]} ]; then
      get_input_result="${mock_input_array[$mock_input_array_index]}"
      # bashsupport disable=BP2001
      # shellcheck disable=SC2206
      ((mock_input_array_index++))
      echo -n "$get_input_result"
    else
      echo "MOCK_INPUT_ARRAY has no more elements" 1>&2
      return 1
    fi
  fi

  return 0
}

###############################################################################
# function yes_no_input
#
# This function requests a yes/no answer from the user, following to a prompt.
#
# Parameters:
#       $1 - The prompt.
#
# Returns:
#       0 - If yes.
#       1 - If no.
###############################################################################
function yes_no_input() {
  if [ -z "$1" ]; then
    echo "Internal error yes_no_input"
    return 1
  fi

  local index=0
  local response="N"
  while [[ $index -lt 10 ]]; do
    index=$((index + 1))
    echo -n "$1"
    if ! get_input; then
      return 1
    fi
```

```
    response=$(echo "$get_input_result" | tr '[:upper:]' '[:lower:]')
    if [ "$response" = "y" ] || [ "$response" = "n" ]; then
      break
    else
      echo -e "\nPlease enter or 'y' or 'n'."
    fi
  done

  echo

  if [ "$response" = "y" ]; then
    return 0
  else
    return 1
  fi
}

################################################################################
# function integer_input
#
# This function prompts the user to enter an integer within a specified range
# and validates the input.
#
# Parameters:
#       $1 - The prompt message to display to the user.
#       $2 - The minimum value of the accepted range.
#       $3 - The maximum value of the accepted range.
#
# Returns:
#       The valid integer input from the user.
#       If the input is invalid or out of range, the function will continue
#       prompting the user until a valid input is provided.
################################################################################
function integer_input() {
  local prompt="$1"
  local min_value="$2"
  local max_value="$3"
  local input=""

  while true; do
    # Display the prompt message and wait for user input
    echo -n "$prompt"

    if ! get_input; then
```

```
      return 1
    fi

    input="$get_input_result"

    # Check if the input is a valid integer
    if [[ "$input" =~ ^-?[0-9]+$ ]]; then
      # Check if the input is within the specified range
      if ((input >= min_value && input <= max_value)); then
        return 0
      else
        echo "Error: Input, $input, must be between $min_value and $max_value."
      fi
    else
      echo "Error: Invalid input- $input. Please enter an integer."
    fi
  done
}
################################################################################
# function new_line_and_tab_to_list
#
# This function takes a string input containing newlines and tabs, and
# converts it into a list (array) of elements.
#
# Parameters:
#       $1 - The input string containing newlines and tabs.
#
# Returns:
#       The resulting list (array) is stored in the global variable
#       'list_result'.
################################################################################
function new_line_and_tab_to_list() {
  local input=$1
  export list_result

  list_result=()
  mapfile -t lines <<<"$input"
  local line
  for line in "${lines[@]}"; do
    IFS=$'\t' read -ra parameters <<<"$line"
    list_result+=("${parameters[@]}")
  done
}
```

```
###############################################################################
# function echo_repeat
#
# This function prints a string 'n' times to stdout.
#
# Parameters:
#       $1 - The string.
#       $2 - Number of times to print the string.
#
# Outputs:
#   String 'n' times to stdout.
#
# Returns:
#       0
###############################################################################
function echo_repeat() {
  local end=$2
  for ((i = 0; i < end; i++)); do
    echo -n "$1"
  done
  echo
}
```

Fonctions DynamoDB utilisées dans ce scénario.

```
###############################################################################
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#       -f file_path - File to store the key pair.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_create_keypair() {
  local key_pair_name file_path response
```

```
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_create_keypair"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
RSA key pair"
    echo " and writes it to a file."
    echo "  -n key_pair_name - A key pair name."
    echo "  -f file_path - File to store the key pair."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:f:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      f) file_path="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
  fi

  response=$(aws ec2 create-key-pair \
```

```
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  }

  if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
  fi

  return 0
}

###############################################################################
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs.
#
# Parameters:
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_key_pairs() {
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_key_pairs"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
```

```
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws ec2 describe-key-pairs \
    --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

################################################################################
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -n security_group_name - The name of the security group.
#       -d security_group_description - The description of the security group.
#
# Returns:
#       The ID of the created security group, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
```

```
function ec2_create_security_group() {
  local security_group_name security_group_description response

  # Function to display usage information
  function usage() {
    echo "function ec2_create_security_group"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -n security_group_name - The name of the security group."
    echo "  -d security_group_description - The description of the security group."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "n:d:h" option; do
    case "${option}" in
      n) security_group_name="${OPTARG}" ;;
      d) security_group_description="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$security_group_name" ]]; then
    errecho "ERROR: You must provide a security group name with the -n parameter."
    return 1
  fi

  if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
 parameter."
    return 1
  fi

  # Create the security group
  response=$(aws ec2 create-security-group \
```

```
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

################################################################################
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 security groups.
#
# Parameters:
#       -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function ec2_describe_security_groups() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_security_groups"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
 groups."
    echo "  -g security_group_id - The ID of the security group to describe
 (optional)."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:h" option; do
    case "${option}" in
```

```
      g) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

  if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
  else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
 (Amazon EC2) security group.
#
# Parameters:
```

```
#       -g security_group_id - The ID of the security group.
#       -i ip_address - The IP address or CIDR block to authorize.
#       -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#       -f from_port - The start of the port range to authorize.
#       -t to_port - The end of the port range to authorize.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_authorize_security_group_ingress() {
  local security_group_id ip_address protocol from_port to_port response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_authorize_security_group_ingress"
    echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
 EC2) security group."
    echo "  -g security_group_id - The ID of the security group."
    echo "  -i ip_address - The IP address or CIDR block to authorize."
    echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
    echo "  -f from_port - The start of the port range to authorize."
    echo "  -t to_port - The end of the port range to authorize."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:i:p:f:t:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      i) ip_address="${OPTARG}" ;;
      p) protocol="${OPTARG}" ;;
      f) from_port="${OPTARG}" ;;
      t) to_port="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
```

```
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -g parameter."
    usage
    return 1
  fi

  if [[ -z "$ip_address" ]]; then
    errecho "ERROR: You must provide an IP address or CIDR block with the -i
 parameter."
    usage
    return 1
  fi

  if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
  fi

  if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
  fi

  if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
  fi

  response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
```

```
    return 1
  }

  return 0
}

###############################################################################
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 images.
#
# Parameters:
#       -i image_ids - A space-separated  list of image IDs (optional).
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_images() {
  local image_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) image_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
```

```
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
  fi

  response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

################################################################################
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE         Comma-separated list of instance types (e.g.,
 t2.micro)
# -h, --help                       Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
################################################################################
function ec2_describe_instance_types() {
  local architecture=""
  local instance_types=""
```

```
  # bashsupport disable=BP5008
  function usage() {
    echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
    echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
 (e.g., x86_64)"
    echo "  -t, --type INSTANCE_TYPE         Comma-separated list of instance types
 (e.g., t2.micro)"
    echo "  -h, --help                       Show this help message"
  }

  while [[ $# -gt 0 ]]; do
    case "$1" in
      -a | --architecture)
        architecture="$2"
        shift 2
        ;;
      -t | --type)
        instance_types="$2"
        shift 2
        ;;
      -h | --help)
        usage
        return 0
        ;;
      *)
        echo "Unknown argument: $1"
        return 1
        ;;
    esac
  done

  if [[ -z "$architecture" ]]; then
    errecho "Error: Architecture not specified."
    usage
    return 1
  fi

  if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
  fi
```

```
local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done
echo -n ']},
  {
  "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=',' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done

echo -n ']}]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
  --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  echo "ERROR: AWS reports describe-instance-types operation failed."
```

```
    return 1
  fi

  echo "$response"
  return 0
}

################################################################################
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#       -t instance_type - The instance type to use (e.g., t2.micro).
#       -k key_pair_name - The name of the key pair to use.
#       -s security_group_id - The ID of the security group to use.
#       -c count - The number of instances to launch (default: 1).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
    echo "  -k key_pair_name - The name of the key pair to use."
    echo "  -s security_group_id - The ID of the security group to use."
    echo "  -c count - The number of instances to launch (default: 1)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:t:k:s:c:h" option; do
```

```
    case "${option}" in
      i) image_id="${OPTARG}" ;;
      t) instance_type="${OPTARG}" ;;
      k) key_pair_name="${OPTARG}" ;;
      s) security_group_id="${OPTARG}" ;;
      c) count="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage
    return 1
  fi

  if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
  fi

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
  fi

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
  fi
```

```
  if [[ -z "$count" ]]; then
    count=1
  fi

  response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

###############################################################################
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID of the instance to describe (optional).
#       -q query - The query to filter the response (optional).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_instances() {
  local instance_id query response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_instances"
```

```
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_id - The ID of the instance to describe (optional)."
    echo "  -q query - The query to filter the response (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:q:h" option; do
    case "${option}" in
      i) instance_id="${OPTARG}" ;;
      q) query="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
  fi

  local query_arg=""
  if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
  else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
```

```
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

###############################################################################
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_stop_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_stop_instances"
    echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
```

```
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 stop-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
  }

  return 0
}

###############################################################################
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
```

```
function ec2_start_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
  }
```

```
  return 0
}

################################################################################
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
 Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#       -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard').
#
# Returns:
#       The allocated Elastic IP address, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_allocate_address() {
  local domain response

  # Function to display usage information
  function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
 (Amazon EC2) instances in a specific AWS Region."
    echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard')."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "d:h" option; do
    case "${option}" in
      d) domain="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
 'standard')."
    return 1
  fi

  if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
  fi

  # Allocate the Elastic IP address
  response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

###############################################################################
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to associate.
```

```
#       -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 associate."
    echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:i:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      i) instance_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
```

```
    return 1
  fi

  if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
  fi

  # Associate the Elastic IP address
  response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

################################################################################
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a association_id - The association ID that represents the association of
 the Elastic IP address with an instance.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_disassociate_address() {
  local association_id response

  # Function to display usage information
  function usage() {
```

```
    echo "function ec2_disassociate_address"
    echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a association_id - The association ID that represents the association
 of the Elastic IP address with an instance."
    echo ""
 }

 # Parse the command-line arguments
 while getopts "a:h" option; do
   case "${option}" in
     a) association_id="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 # Validate the input parameters
 if [[ -z "$association_id" ]]; then
   errecho "ERROR: You must provide an association ID with the -a parameter."
   return 1
 fi

 response=$(aws ec2 disassociate-address \
   --association-id "$association_id") || {
   aws_cli_error_log ${?}
   errecho "ERROR: AWS reports disassociate-address operation failed."
   errecho "$response"
   return 1
 }

 return 0
}

################################################################################
# function ec2_release_address
```

```
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_release_address() {
  local allocation_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_release_address"
    echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 release."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
```

```
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 release-address \
    --allocation-id "$allocation_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}

###############################################################################
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#       -i instance_ids - A space-separated list of instance IDs.
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_terminate_instances() {
  local instance_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_terminate_instances"
    echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_ids - A space-separated list of instance IDs."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
```

```
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Check if instance ID is provided
  if [[ -z "${instance_ids}" ]]; then
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
    --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
  }

  return 0
}

################################################################################
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -i security_group_id - The ID of the security group to delete.
#
```

```
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_security_group() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_security_group"
    echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -i security_group_id - The ID of the security group to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
 text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
```

```
  }

  return 0
}

################################################################################
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#
# And:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function ec2_delete_keypair() {
  local key_pair_name response

  local option OPTARG # Required to use getopts command in a function.
  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_keypair"
    echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
    echo "  -n key_pair_name - A key pair name."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans ce scénario.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
```

```
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AllocateAddress
  - AssociateAddress
  - AuthorizeSecurityGroupIngress
  - CreateKeyPair
  - CreateSecurityGroup
  - DeleteKeyPair
  - DeleteSecurityGroup
  - DescribeImages
  - DescribeInstanceTypes
  - DescribeInstances
  - DescribeKeyPairs
  - DescribeSecurityGroups
  - DisassociateAddress

- ReleaseAddress
- RunInstances
- StartInstances
- StopInstances
- TerminateInstances
- UnmonitorInstances

Actions

`AllocateAddress`

L'exemple de code suivant montre comment utiliser`AllocateAddress`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
 Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#       -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard').
#
# Returns:
#       The allocated Elastic IP address, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
```

```
function ec2_allocate_address() {
  local domain response

  # Function to display usage information
  function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
 (Amazon EC2) instances in a specific AWS Region."
    echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard')."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "d:h" option; do
    case "${option}" in
      d) domain="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
 'standard')."
    return 1
  fi

  if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
  fi

  # Allocate the Elastic IP address
  response=$(aws ec2 allocate-address \
```

```
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
```

```
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous AllocateAddressà la section Référence des AWS CLI commandes.

`AssociateAddress`

L'exemple de code suivant montre comment utiliser`AssociateAddress`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to associate.
```

```
#       -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 associate."
    echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:i:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      i) instance_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
```

```
    return 1
  fi

  if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
  fi

  # Associate the Elastic IP address
  response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
```

```
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateAddressà la section Référence des AWS CLI commandes.

`AuthorizeSecurityGroupIngress`

L'exemple de code suivant montre comment utiliser`AuthorizeSecurityGroupIngress`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
 (Amazon EC2) security group.
#
# Parameters:
#       -g security_group_id - The ID of the security group.
#       -i ip_address - The IP address or CIDR block to authorize.
#       -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#       -f from_port - The start of the port range to authorize.
#       -t to_port - The end of the port range to authorize.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_authorize_security_group_ingress() {
  local security_group_id ip_address protocol from_port to_port response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_authorize_security_group_ingress"
    echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
 EC2) security group."
    echo "  -g security_group_id - The ID of the security group."
    echo "  -i ip_address - The IP address or CIDR block to authorize."
    echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
    echo "  -f from_port - The start of the port range to authorize."
    echo "  -t to_port - The end of the port range to authorize."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:i:p:f:t:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      i) ip_address="${OPTARG}" ;;
      p) protocol="${OPTARG}" ;;
      f) from_port="${OPTARG}" ;;
      t) to_port="${OPTARG}" ;;
      h)
```

```
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
  errecho "ERROR: You must provide a security group ID with the -g parameter."
  usage
  return 1
fi

if [[ -z "$ip_address" ]]; then
  errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
  usage
  return 1
fi

if [[ -z "$protocol" ]]; then
  errecho "ERROR: You must provide a protocol with the -p parameter."
  usage
  return 1
fi

if [[ -z "$from_port" ]]; then
  errecho "ERROR: You must provide a start port with the -f parameter."
  usage
  return 1
fi

if [[ -z "$to_port" ]]; then
  errecho "ERROR: You must provide an end port with the -t parameter."
  usage
  return 1
fi

response=$(aws ec2 authorize-security-group-ingress \
```

```
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
```

```
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous AuthorizeSecurityGroupIngressà la section Référence des AWS CLI commandes.

`CreateKeyPair`

L'exemple de code suivant montre comment utiliser`CreateKeyPair`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
```

```
#       -n key_pair_name - A key pair name.
#       -f file_path - File to store the key pair.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_create_keypair() {
  local key_pair_name file_path response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_create_keypair"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
 RSA key pair"
    echo " and writes it to a file."
    echo "  -n key_pair_name - A key pair name."
    echo "  -f file_path - File to store the key pair."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:f:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      f) file_path="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
```

```
  fi

  if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
  fi

  response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  }

  if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
  fi

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
```

```
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateKeyPairà la section Référence des AWS CLI commandes.

`CreateSecurityGroup`

L'exemple de code suivant montre comment utiliser`CreateSecurityGroup`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -n security_group_name - The name of the security group.
#       -d security_group_description - The description of the security group.
#
# Returns:
#       The ID of the created security group, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_create_security_group() {
  local security_group_name security_group_description response

  # Function to display usage information
  function usage() {
    echo "function ec2_create_security_group"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -n security_group_name - The name of the security group."
    echo "  -d security_group_description - The description of the security group."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "n:d:h" option; do
    case "${option}" in
      n) security_group_name="${OPTARG}" ;;
      d) security_group_description="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
```

```
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$security_group_name" ]]; then
    errecho "ERROR: You must provide a security group name with the -n parameter."
    return 1
  fi

  if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
 parameter."
    return 1
  fi

  # Create the security group
  response=$(aws ec2 create-security-group \
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSecurityGroupà la section Référence des AWS CLI commandes.

`DeleteKeyPair`

L'exemple de code suivant montre comment utiliser`DeleteKeyPair`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function ec2_delete_keypair() {
  local key_pair_name response

  local option OPTARG # Required to use getopts command in a function.
  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_keypair"
    echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
    echo "  -n key_pair_name - A key pair name."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
```

```
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteKeyPairà la section Référence des AWS CLI commandes.

`DeleteSecurityGroup`

L'exemple de code suivant montre comment utiliser`DeleteSecurityGroup`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -i security_group_id - The ID of the security group to delete.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_security_group() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_security_group"
    echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -i security_group_id - The ID of the security group to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
```

```
    usage
    return 1
  fi

  response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
 text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
```

```
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteSecurityGroupà la section Référence des AWS CLI commandes.

`DescribeImages`

L'exemple de code suivant montre comment utiliser`DescribeImages`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 images.
#
# Parameters:
#       -i image_ids - A space-separated  list of image IDs (optional).
```

```
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_images() {
  local image_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) image_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
  fi

  response=$(aws ec2 describe-images \
```

```
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
```

```
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeImagesà la section Référence des AWS CLI commandes.

`DescribeInstanceTypes`

L'exemple de code suivant montre comment utiliser`DescribeInstanceTypes`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE         Comma-separated list of instance types (e.g.,
 t2.micro)
```

```
# -h, --help                      Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#############################################################################
function ec2_describe_instance_types() {
  local architecture=""
  local instance_types=""

  # bashsupport disable=BP5008
  function usage() {
    echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
    echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
 (e.g., x86_64)"
    echo "  -t, --type INSTANCE_TYPE         Comma-separated list of instance types
 (e.g., t2.micro)"
    echo "  -h, --help                       Show this help message"
  }

  while [[ $# -gt 0 ]]; do
    case "$1" in
      -a | --architecture)
        architecture="$2"
        shift 2
        ;;
      -t | --type)
        instance_types="$2"
        shift 2
        ;;
      -h | --help)
        usage
        return 0
        ;;
      *)
        echo "Unknown argument: $1"
        return 1
        ;;
    esac
  done

  if [[ -z "$architecture" ]]; then
    errecho "Error: Architecture not specified."
    usage
```

```
  return 1
fi

if [[ -z "$instance_types" ]]; then
  errecho "Error: Instance type not specified."
  usage
  return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done
echo -n ']},
  {
  "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=',' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done

echo -n ']}]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
  --query 'InstanceTypes[*].[InstanceType]' --output text)
```

```
  local error_code=$?

  rm "$tmp_json_file"

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
  fi

  echo "$response"
  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
```

```
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstanceTypesà la section Référence des AWS CLI commandes.

`DescribeInstances`

L'exemple de code suivant montre comment utiliser`DescribeInstances`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
```

```
#       -i instance_id - The ID of the instance to describe (optional).
#       -q query - The query to filter the response (optional).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_instances() {
  local instance_id query response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_instances"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_id - The ID of the instance to describe (optional)."
    echo "  -q query - The query to filter the response (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:q:h" option; do
    case "${option}" in
      i) instance_id="${OPTARG}" ;;
      q) query="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$instance_id" ]]; then
```

```
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
  fi

  local query_arg=""
  if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
  else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
```

```
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeInstancesà la section Référence des AWS CLI commandes.

`DescribeKeyPairs`

L'exemple de code suivant montre comment utiliser`DescribeKeyPairs`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs.
#
# Parameters:
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_key_pairs() {
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_key_pairs"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws ec2 describe-key-pairs \
    --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
```

```
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeKeyPairsà la section Référence des AWS CLI commandes.

`DescribeSecurityGroups`

L'exemple de code suivant montre comment utiliser`DescribeSecurityGroups`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 security groups.
#
# Parameters:
#       -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_security_groups() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_security_groups"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
 groups."
    echo "  -g security_group_id - The ID of the security group to describe
 (optional)."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

  if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
  else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
```

```
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSecurityGroupsà la section Référence des AWS CLI commandes.

`DisassociateAddress`

L'exemple de code suivant montre comment utiliser`DisassociateAddress`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a association_id - The association ID that represents the association of
 the Elastic IP address with an instance.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_disassociate_address() {
  local association_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_disassociate_address"
    echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a association_id - The association ID that represents the association
 of the Elastic IP address with an instance."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) association_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$association_id" ]]; then
    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 disassociate-address \
    --association-id "$association_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
```

```
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateAddressà la section Référence des AWS CLI commandes.

`ReleaseAddress`

L'exemple de code suivant montre comment utiliser`ReleaseAddress`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function ec2_release_address
#
```

```
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_release_address() {
  local allocation_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_release_address"
    echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 release."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
```

```
    return 1
  fi

  response=$(aws ec2 release-address \
    --allocation-id "$allocation_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
```

```
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous ReleaseAddressà la section Référence des AWS CLI commandes.

`RunInstances`

L'exemple de code suivant montre comment utiliser`RunInstances`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i image_id - The ID of the Amazon Machine Image (AMI) to use.
```

```
#       -t instance_type - The instance type to use (e.g., t2.micro).
#       -k key_pair_name - The name of the key pair to use.
#       -s security_group_id - The ID of the security group to use.
#       -c count - The number of instances to launch (default: 1).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
    echo "  -k key_pair_name - The name of the key pair to use."
    echo "  -s security_group_id - The ID of the security group to use."
    echo "  -c count - The number of instances to launch (default: 1)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:t:k:s:c:h" option; do
    case "${option}" in
      i) image_id="${OPTARG}" ;;
      t) instance_type="${OPTARG}" ;;
      k) key_pair_name="${OPTARG}" ;;
      s) security_group_id="${OPTARG}" ;;
      c) count="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
```

```
    esac
  done
  export OPTIND=1

  if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage
    return 1
  fi

  if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
  fi

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
  fi

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
  fi

  if [[ -z "$count" ]]; then
    count=1
  fi

  response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
```

```
  }

  echo "$response"

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
```

```
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous RunInstancesà la section Référence des AWS CLI commandes.

`StartInstances`

L'exemple de code suivant montre comment utiliser`StartInstances`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_start_instances() {
```

```
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
  }
```

```
  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#####################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#####################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
```

```
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous StartInstancesà la section Référence des AWS CLI commandes.

`StopInstances`

L'exemple de code suivant montre comment utiliser`StopInstances`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function ec2_stop_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
```

```
  function usage() {
    echo "function ec2_stop_instances"
    echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 stop-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#####################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#####################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous StopInstancesà la section Référence des AWS CLI commandes.

`TerminateInstances`

L'exemple de code suivant montre comment utiliser`TerminateInstances`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#       -i instance_ids - A space-separated list of instance IDs.
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_terminate_instances() {
  local instance_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_terminate_instances"
    echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_ids - A space-separated list of instance IDs."
```

```
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Check if instance ID is provided
  if [[ -z "${instance_ids}" ]]; then
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
    --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
  }

  return 0
}
```

Fonctions utilitaires utilisées dans cet exemple.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous TerminateInstancesà la section Référence des AWS CLI commandes.

Scénarios

Création d'un VPC avec des sous-réseaux privés et des passerelles NAT

L'exemple de code suivant illustre comment :

- Créez un VPC avec des sous-réseaux privés et des passerelles NAT à l'aide de la CLI.
- Configurez les composants nécessaires, notamment le VPC, les sous-réseaux, les tables de routage et les passerelles NAT.
- Configurez les groupes de sécurité et les rôles IAM pour un accès et une sécurité appropriés.
- Utilisez les commandes CLI pour automatiser la création et la configuration de ces ressources.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# VPC with Private Subnets and NAT Gateways (IMDSv2 Compliant Version)
# This script creates a VPC with public and private subnets in two Availability
 Zones,
# NAT gateways, an internet gateway, route tables, a VPC endpoint for S3,
# security groups, a launch template, an Auto Scaling group, and an Application Load
 Balancer.

# Set up logging
LOG_FILE="vpc-private-subnets-nat.log"
exec > >(tee -a "$LOG_FILE") 2>&1

# Cleanup function to delete all created resources
cleanup_resources() {
```

```
 echo "Cleaning up resources..."

 # Delete Auto Scaling group if it exists
 if [ -n "${ASG_NAME:-}" ]; then
   echo "Deleting Auto Scaling group: $ASG_NAME"
   aws autoscaling delete-auto-scaling-group --auto-scaling-group-name "$ASG_NAME"
--force-delete
   echo "Waiting for Auto Scaling group to be deleted..."
   aws autoscaling wait auto-scaling-groups-deleted --auto-scaling-group-names
"$ASG_NAME"
 fi

 # Delete load balancer if it exists
 if [ -n "${LB_ARN:-}" ]; then
   echo "Deleting load balancer: $LB_ARN"
   aws elbv2 delete-load-balancer --load-balancer-arn "$LB_ARN"
   # Wait for load balancer to be deleted
   sleep 30
 fi

 # Delete target group if it exists
 if [ -n "${TARGET_GROUP_ARN:-}" ]; then
   echo "Deleting target group: $TARGET_GROUP_ARN"
   aws elbv2 delete-target-group --target-group-arn "$TARGET_GROUP_ARN"
 fi

 # Delete launch template if it exists
 if [ -n "${LAUNCH_TEMPLATE_NAME:-}" ]; then
   echo "Deleting launch template: $LAUNCH_TEMPLATE_NAME"
   aws ec2 delete-launch-template --launch-template-name "$LAUNCH_TEMPLATE_NAME"
 fi

 # Delete NAT Gateways if they exist
 if [ -n "${NAT_GW1_ID:-}" ]; then
   echo "Deleting NAT Gateway 1: $NAT_GW1_ID"
   aws ec2 delete-nat-gateway --nat-gateway-id "$NAT_GW1_ID"
 fi

 if [ -n "${NAT_GW2_ID:-}" ]; then
   echo "Deleting NAT Gateway 2: $NAT_GW2_ID"
   aws ec2 delete-nat-gateway --nat-gateway-id "$NAT_GW2_ID"
 fi

 # Wait for NAT Gateways to be deleted
```

```
  if [ -n "${NAT_GW1_ID:-}" ] || [ -n "${NAT_GW2_ID:-}" ]; then
    echo "Waiting for NAT Gateways to be deleted..."
    sleep 60
  fi

  # Release Elastic IPs if they exist
  if [ -n "${EIP1_ALLOC_ID:-}" ]; then
    echo "Releasing Elastic IP 1: $EIP1_ALLOC_ID"
    aws ec2 release-address --allocation-id "$EIP1_ALLOC_ID"
  fi

  if [ -n "${EIP2_ALLOC_ID:-}" ]; then
    echo "Releasing Elastic IP 2: $EIP2_ALLOC_ID"
    aws ec2 release-address --allocation-id "$EIP2_ALLOC_ID"
  fi

  # Delete VPC endpoint if it exists
  if [ -n "${VPC_ENDPOINT_ID:-}" ]; then
    echo "Deleting VPC endpoint: $VPC_ENDPOINT_ID"
    aws ec2 delete-vpc-endpoints --vpc-endpoint-ids "$VPC_ENDPOINT_ID"
  fi

  # Delete security groups if they exist
  if [ -n "${APP_SG_ID:-}" ]; then
    echo "Deleting application security group: $APP_SG_ID"
    aws ec2 delete-security-group --group-id "$APP_SG_ID"
  fi

  if [ -n "${LB_SG_ID:-}" ]; then
    echo "Deleting load balancer security group: $LB_SG_ID"
    aws ec2 delete-security-group --group-id "$LB_SG_ID"
  fi

  # Detach and delete Internet Gateway if it exists
  if [ -n "${IGW_ID:-}" ] && [ -n "${VPC_ID:-}" ]; then
    echo "Detaching Internet Gateway: $IGW_ID from VPC: $VPC_ID"
    aws ec2 detach-internet-gateway --internet-gateway-id "$IGW_ID" --vpc-id
"$VPC_ID"
    echo "Deleting Internet Gateway: $IGW_ID"
    aws ec2 delete-internet-gateway --internet-gateway-id "$IGW_ID"
  fi

  # Delete route table associations and route tables if they exist
  if [ -n "${PUBLIC_RT_ASSOC1_ID:-}" ]; then
```

```
  echo "Disassociating public route table from subnet 1: $PUBLIC_RT_ASSOC1_ID"
  aws ec2 disassociate-route-table --association-id "$PUBLIC_RT_ASSOC1_ID"
fi

if [ -n "${PUBLIC_RT_ASSOC2_ID:-}" ]; then
  echo "Disassociating public route table from subnet 2: $PUBLIC_RT_ASSOC2_ID"
  aws ec2 disassociate-route-table --association-id "$PUBLIC_RT_ASSOC2_ID"
fi

if [ -n "${PRIVATE_RT1_ASSOC_ID:-}" ]; then
  echo "Disassociating private route table 1: $PRIVATE_RT1_ASSOC_ID"
  aws ec2 disassociate-route-table --association-id "$PRIVATE_RT1_ASSOC_ID"
fi

if [ -n "${PRIVATE_RT2_ASSOC_ID:-}" ]; then
  echo "Disassociating private route table 2: $PRIVATE_RT2_ASSOC_ID"
  aws ec2 disassociate-route-table --association-id "$PRIVATE_RT2_ASSOC_ID"
fi

if [ -n "${PUBLIC_RT_ID:-}" ]; then
  echo "Deleting public route table: $PUBLIC_RT_ID"
  aws ec2 delete-route-table --route-table-id "$PUBLIC_RT_ID"
fi

if [ -n "${PRIVATE_RT1_ID:-}" ]; then
  echo "Deleting private route table 1: $PRIVATE_RT1_ID"
  aws ec2 delete-route-table --route-table-id "$PRIVATE_RT1_ID"
fi

if [ -n "${PRIVATE_RT2_ID:-}" ]; then
  echo "Deleting private route table 2: $PRIVATE_RT2_ID"
  aws ec2 delete-route-table --route-table-id "$PRIVATE_RT2_ID"
fi

# Delete subnets if they exist
if [ -n "${PUBLIC_SUBNET1_ID:-}" ]; then
  echo "Deleting public subnet 1: $PUBLIC_SUBNET1_ID"
  aws ec2 delete-subnet --subnet-id "$PUBLIC_SUBNET1_ID"
fi

if [ -n "${PUBLIC_SUBNET2_ID:-}" ]; then
  echo "Deleting public subnet 2: $PUBLIC_SUBNET2_ID"
  aws ec2 delete-subnet --subnet-id "$PUBLIC_SUBNET2_ID"
fi
```

```
  if [ -n "${PRIVATE_SUBNET1_ID:-}" ]; then
    echo "Deleting private subnet 1: $PRIVATE_SUBNET1_ID"
    aws ec2 delete-subnet --subnet-id "$PRIVATE_SUBNET1_ID"
  fi

  if [ -n "${PRIVATE_SUBNET2_ID:-}" ]; then
    echo "Deleting private subnet 2: $PRIVATE_SUBNET2_ID"
    aws ec2 delete-subnet --subnet-id "$PRIVATE_SUBNET2_ID"
  fi

  # Delete VPC if it exists
  if [ -n "${VPC_ID:-}" ]; then
    echo "Deleting VPC: $VPC_ID"
    aws ec2 delete-vpc --vpc-id "$VPC_ID"
  fi

  echo "Cleanup completed."
}

# Error handling function
handle_error() {
  echo "ERROR: $1"
  echo "Attempting to clean up resources..."
  cleanup_resources
  exit 1
}

# Function to check command success
check_command() {
  if [ $? -ne 0 ]; then
    handle_error "$1"
  fi
}

# Generate a random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
echo "Using random identifier: $RANDOM_ID"

# Create VPC
echo "Creating VPC..."
VPC_RESULT=$(aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 "ResourceType=vpc,Tags=[{Key=Name,Value=ProductionVPC-$RANDOM_ID}]")
check_command "Failed to create VPC"
```

```
VPC_ID=$(echo "$VPC_RESULT" | jq -r '.Vpc.VpcId')
echo "VPC created with ID: $VPC_ID"

# Get Availability Zones
echo "Getting Availability Zones..."
AZ_RESULT=$(aws ec2 describe-availability-zones --query
 'AvailabilityZones[0:2].ZoneName' --output text)
check_command "Failed to get Availability Zones"

# Convert space-separated output to array
read -r -a AZS <<< "$AZ_RESULT"
AZ1=${AZS[0]}
AZ2=${AZS[1]}
echo "Using Availability Zones: $AZ1 and $AZ2"

# Create subnets
echo "Creating subnets..."
PUBLIC_SUBNET1_RESULT=$(aws ec2 create-subnet --vpc-id "$VPC_ID" --
cidr-block 10.0.0.0/24 --availability-zone "$AZ1" --tag-specifications
 "ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet1-$RANDOM_ID}]")
check_command "Failed to create public subnet 1"
PUBLIC_SUBNET1_ID=$(echo "$PUBLIC_SUBNET1_RESULT" | jq -r '.Subnet.SubnetId')

PRIVATE_SUBNET1_RESULT=$(aws ec2 create-subnet --vpc-id "$VPC_ID" --
cidr-block 10.0.1.0/24 --availability-zone "$AZ1" --tag-specifications
 "ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet1-$RANDOM_ID}]")
check_command "Failed to create private subnet 1"
PRIVATE_SUBNET1_ID=$(echo "$PRIVATE_SUBNET1_RESULT" | jq -r '.Subnet.SubnetId')

PUBLIC_SUBNET2_RESULT=$(aws ec2 create-subnet --vpc-id "$VPC_ID" --
cidr-block 10.0.2.0/24 --availability-zone "$AZ2" --tag-specifications
 "ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet2-$RANDOM_ID}]")
check_command "Failed to create public subnet 2"
PUBLIC_SUBNET2_ID=$(echo "$PUBLIC_SUBNET2_RESULT" | jq -r '.Subnet.SubnetId')

PRIVATE_SUBNET2_RESULT=$(aws ec2 create-subnet --vpc-id "$VPC_ID" --
cidr-block 10.0.3.0/24 --availability-zone "$AZ2" --tag-specifications
 "ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet2-$RANDOM_ID}]")
check_command "Failed to create private subnet 2"
PRIVATE_SUBNET2_ID=$(echo "$PRIVATE_SUBNET2_RESULT" | jq -r '.Subnet.SubnetId')

echo "Subnets created with IDs:"
echo "Public Subnet 1: $PUBLIC_SUBNET1_ID"
```

```
echo "Private Subnet 1: $PRIVATE_SUBNET1_ID"
echo "Public Subnet 2: $PUBLIC_SUBNET2_ID"
echo "Private Subnet 2: $PRIVATE_SUBNET2_ID"

# Create Internet Gateway
echo "Creating Internet Gateway..."
IGW_RESULT=$(aws ec2 create-internet-gateway --tag-specifications
 "ResourceType=internet-gateway,Tags=[{Key=Name,Value=ProductionIGW-$RANDOM_ID}]")
check_command "Failed to create Internet Gateway"
IGW_ID=$(echo "$IGW_RESULT" | jq -r '.InternetGateway.InternetGatewayId')
echo "Internet Gateway created with ID: $IGW_ID"

# Attach Internet Gateway to VPC
echo "Attaching Internet Gateway to VPC..."
aws ec2 attach-internet-gateway --internet-gateway-id "$IGW_ID" --vpc-id "$VPC_ID"
check_command "Failed to attach Internet Gateway to VPC"

# Create route tables
echo "Creating route tables..."
PUBLIC_RT_RESULT=$(aws ec2 create-route-table --vpc-id "$VPC_ID" --tag-
specifications "ResourceType=route-table,Tags=[{Key=Name,Value=PublicRouteTable-
$RANDOM_ID}]")
check_command "Failed to create public route table"
PUBLIC_RT_ID=$(echo "$PUBLIC_RT_RESULT" | jq -r '.RouteTable.RouteTableId')

PRIVATE_RT1_RESULT=$(aws ec2 create-route-table --vpc-id "$VPC_ID" --tag-
specifications "ResourceType=route-table,Tags=[{Key=Name,Value=PrivateRouteTable1-
$RANDOM_ID}]")
check_command "Failed to create private route table 1"
PRIVATE_RT1_ID=$(echo "$PRIVATE_RT1_RESULT" | jq -r '.RouteTable.RouteTableId')

PRIVATE_RT2_RESULT=$(aws ec2 create-route-table --vpc-id "$VPC_ID" --tag-
specifications "ResourceType=route-table,Tags=[{Key=Name,Value=PrivateRouteTable2-
$RANDOM_ID}]")
check_command "Failed to create private route table 2"
PRIVATE_RT2_ID=$(echo "$PRIVATE_RT2_RESULT" | jq -r '.RouteTable.RouteTableId')

echo "Route tables created with IDs:"
echo "Public Route Table: $PUBLIC_RT_ID"
echo "Private Route Table 1: $PRIVATE_RT1_ID"
echo "Private Route Table 2: $PRIVATE_RT2_ID"

# Add route to Internet Gateway in public route table
echo "Adding route to Internet Gateway in public route table..."
```

```
aws ec2 create-route --route-table-id "$PUBLIC_RT_ID" --destination-cidr-block
 0.0.0.0/0 --gateway-id "$IGW_ID"
check_command "Failed to add route to Internet Gateway"

# Associate subnets with route tables
echo "Associating subnets with route tables..."
PUBLIC_RT_ASSOC1_RESULT=$(aws ec2 associate-route-table --route-table-id
 "$PUBLIC_RT_ID" --subnet-id "$PUBLIC_SUBNET1_ID")
check_command "Failed to associate public subnet 1 with route table"
PUBLIC_RT_ASSOC1_ID=$(echo "$PUBLIC_RT_ASSOC1_RESULT" | jq -r '.AssociationId')

PUBLIC_RT_ASSOC2_RESULT=$(aws ec2 associate-route-table --route-table-id
 "$PUBLIC_RT_ID" --subnet-id "$PUBLIC_SUBNET2_ID")
check_command "Failed to associate public subnet 2 with route table"
PUBLIC_RT_ASSOC2_ID=$(echo "$PUBLIC_RT_ASSOC2_RESULT" | jq -r '.AssociationId')

PRIVATE_RT1_ASSOC_RESULT=$(aws ec2 associate-route-table --route-table-id
 "$PRIVATE_RT1_ID" --subnet-id "$PRIVATE_SUBNET1_ID")
check_command "Failed to associate private subnet 1 with route table"
PRIVATE_RT1_ASSOC_ID=$(echo "$PRIVATE_RT1_ASSOC_RESULT" | jq -r '.AssociationId')

PRIVATE_RT2_ASSOC_RESULT=$(aws ec2 associate-route-table --route-table-id
 "$PRIVATE_RT2_ID" --subnet-id "$PRIVATE_SUBNET2_ID")
check_command "Failed to associate private subnet 2 with route table"
PRIVATE_RT2_ASSOC_ID=$(echo "$PRIVATE_RT2_ASSOC_RESULT" | jq -r '.AssociationId')

echo "Route table associations created with IDs:"
echo "Public Subnet 1 Association: $PUBLIC_RT_ASSOC1_ID"
echo "Public Subnet 2 Association: $PUBLIC_RT_ASSOC2_ID"
echo "Private Subnet 1 Association: $PRIVATE_RT1_ASSOC_ID"
echo "Private Subnet 2 Association: $PRIVATE_RT2_ASSOC_ID"

# Create NAT Gateways
echo "Creating NAT Gateways..."

# Allocate Elastic IPs for NAT Gateways
echo "Allocating Elastic IPs for NAT Gateways..."
EIP1_RESULT=$(aws ec2 allocate-address --domain vpc --tag-specifications
 "ResourceType=elastic-ip,Tags=[{Key=Name,Value=NAT1-EIP-$RANDOM_ID}]")
check_command "Failed to allocate Elastic IP 1"
EIP1_ALLOC_ID=$(echo "$EIP1_RESULT" | jq -r '.AllocationId')

EIP2_RESULT=$(aws ec2 allocate-address --domain vpc --tag-specifications
 "ResourceType=elastic-ip,Tags=[{Key=Name,Value=NAT2-EIP-$RANDOM_ID}]")
```

```
check_command "Failed to allocate Elastic IP 2"
EIP2_ALLOC_ID=$(echo "$EIP2_RESULT" | jq -r '.AllocationId')

echo "Elastic IPs allocated with IDs:"
echo "EIP 1 Allocation ID: $EIP1_ALLOC_ID"
echo "EIP 2 Allocation ID: $EIP2_ALLOC_ID"

# Create NAT Gateways
echo "Creating NAT Gateway in public subnet 1..."
NAT_GW1_RESULT=$(aws ec2 create-nat-gateway --subnet-id
 "$PUBLIC_SUBNET1_ID" --allocation-id "$EIP1_ALLOC_ID" --tag-specifications
 "ResourceType=natgateway,Tags=[{Key=Name,Value=NAT-Gateway1-$RANDOM_ID}]")
check_command "Failed to create NAT Gateway 1"
NAT_GW1_ID=$(echo "$NAT_GW1_RESULT" | jq -r '.NatGateway.NatGatewayId')

echo "Creating NAT Gateway in public subnet 2..."
NAT_GW2_RESULT=$(aws ec2 create-nat-gateway --subnet-id
 "$PUBLIC_SUBNET2_ID" --allocation-id "$EIP2_ALLOC_ID" --tag-specifications
 "ResourceType=natgateway,Tags=[{Key=Name,Value=NAT-Gateway2-$RANDOM_ID}]")
check_command "Failed to create NAT Gateway 2"
NAT_GW2_ID=$(echo "$NAT_GW2_RESULT" | jq -r '.NatGateway.NatGatewayId')

echo "NAT Gateways created with IDs:"
echo "NAT Gateway 1: $NAT_GW1_ID"
echo "NAT Gateway 2: $NAT_GW2_ID"

# Wait for NAT Gateways to be available
echo "Waiting for NAT Gateways to be available..."
aws ec2 wait nat-gateway-available --nat-gateway-ids "$NAT_GW1_ID"
check_command "NAT Gateway 1 did not become available"
aws ec2 wait nat-gateway-available --nat-gateway-ids "$NAT_GW2_ID"
check_command "NAT Gateway 2 did not become available"
echo "NAT Gateways are now available"

# Add routes to NAT Gateways in private route tables
echo "Adding routes to NAT Gateways in private route tables..."
aws ec2 create-route --route-table-id "$PRIVATE_RT1_ID" --destination-cidr-block
 0.0.0.0/0 --nat-gateway-id "$NAT_GW1_ID"
check_command "Failed to add route to NAT Gateway 1"

aws ec2 create-route --route-table-id "$PRIVATE_RT2_ID" --destination-cidr-block
 0.0.0.0/0 --nat-gateway-id "$NAT_GW2_ID"
check_command "Failed to add route to NAT Gateway 2"
```

```
# Create VPC Endpoint for S3
echo "Creating VPC Endpoint for S3..."
S3_PREFIX_LIST_ID=$(aws ec2 describe-prefix-lists --filters "Name=prefix-
list-name,Values=com.amazonaws.$(aws configure get region).s3" --query
 'PrefixLists[0].PrefixListId' --output text)
check_command "Failed to get S3 prefix list ID"

VPC_ENDPOINT_RESULT=$(aws ec2 create-vpc-endpoint --vpc-id "$VPC_ID" --
service-name "com.amazonaws.$(aws configure get region).s3" --route-table-ids
 "$PRIVATE_RT1_ID" "$PRIVATE_RT2_ID" --tag-specifications "ResourceType=vpc-
endpoint,Tags=[{Key=Name,Value=S3-Endpoint-$RANDOM_ID}]")
check_command "Failed to create VPC endpoint for S3"
VPC_ENDPOINT_ID=$(echo "$VPC_ENDPOINT_RESULT" | jq -r '.VpcEndpoint.VpcEndpointId')
echo "VPC Endpoint created with ID: $VPC_ENDPOINT_ID"

# Create security groups
echo "Creating security groups..."
LB_SG_RESULT=$(aws ec2 create-security-group --group-name "LoadBalancerSG-
$RANDOM_ID" --description "Security group for the load balancer"
 --vpc-id "$VPC_ID" --tag-specifications "ResourceType=security-
group,Tags=[{Key=Name,Value=LoadBalancerSG-$RANDOM_ID}]")
check_command "Failed to create load balancer security group"
LB_SG_ID=$(echo "$LB_SG_RESULT" | jq -r '.GroupId')

# Allow inbound HTTP traffic from anywhere to the load balancer
aws ec2 authorize-security-group-ingress --group-id "$LB_SG_ID" --protocol tcp --
port 80 --cidr 0.0.0.0/0
check_command "Failed to authorize ingress to load balancer security group"

APP_SG_RESULT=$(aws ec2 create-security-group --group-name "AppServerSG-$RANDOM_ID"
 --description "Security group for the application servers" --vpc-id "$VPC_ID" --
tag-specifications "ResourceType=security-group,Tags=[{Key=Name,Value=AppServerSG-
$RANDOM_ID}]")
check_command "Failed to create application server security group"
APP_SG_ID=$(echo "$APP_SG_RESULT" | jq -r '.GroupId')

# Allow inbound HTTP traffic from the load balancer security group to the
 application servers
aws ec2 authorize-security-group-ingress --group-id "$APP_SG_ID" --protocol tcp --
port 80 --source-group "$LB_SG_ID"
check_command "Failed to authorize ingress to application server security group"

echo "Security groups created with IDs:"
echo "Load Balancer Security Group: $LB_SG_ID"
```

```
echo "Application Server Security Group: $APP_SG_ID"

# Create a launch template
echo "Creating launch template..."

# Create user data script with IMDSv2 support
cat > user-data.sh << 'EOF'
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd

# Use IMDSv2 with session token
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600")
AZ=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/
meta-data/placement/availability-zone)
HOSTNAME=$(hostname -f)

echo "<h1>Hello from $HOSTNAME in $AZ</h1>" > /var/www/html/index.html
EOF

# Encode user data
USER_DATA=$(base64 -w 0 user-data.sh)

# Get latest Amazon Linux 2 AMI
echo "Getting latest Amazon Linux 2 AMI..."
AMI_ID=$(aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-
ami-hvm-*-x86_64-gp2" "Name=state,Values=available" --query 'sort_by(Images,
 &CreationDate)[-1].ImageId' --output text)
check_command "Failed to get latest Amazon Linux 2 AMI"
echo "Using AMI: $AMI_ID"

# Create launch template with IMDSv2 required
LAUNCH_TEMPLATE_NAME="AppServerTemplate-$RANDOM_ID"
echo "Creating launch template: $LAUNCH_TEMPLATE_NAME"

aws ec2 create-launch-template \
  --launch-template-name "$LAUNCH_TEMPLATE_NAME" \
  --version-description "Initial version" \
  --tag-specifications "ResourceType=launch-template,Tags=[{Key=Name,Value=
$LAUNCH_TEMPLATE_NAME}]" \
  --launch-template-data "{
```

```
    \"NetworkInterfaces\": [{
      \"DeviceIndex\": 0,
      \"Groups\": [\"$APP_SG_ID\"],
      \"DeleteOnTermination\": true
    }],
    \"ImageId\": \"$AMI_ID\",
    \"InstanceType\": \"t3.micro\",
    \"UserData\": \"$USER_DATA\",
    \"MetadataOptions\": {
      \"HttpTokens\": \"required\",
      \"HttpEndpoint\": \"enabled\"
    },
    \"TagSpecifications\": [{
      \"ResourceType\": \"instance\",
      \"Tags\": [{
        \"Key\": \"Name\",
        \"Value\": \"AppServer-$RANDOM_ID\"
      }]
    }]
  }"
check_command "Failed to create launch template"

# Create target group
echo "Creating target group..."
TARGET_GROUP_NAME="AppTargetGroup-$RANDOM_ID"
TARGET_GROUP_RESULT=$(aws elbv2 create-target-group \
  --name "$TARGET_GROUP_NAME" \
  --protocol HTTP \
  --port 80 \
  --vpc-id "$VPC_ID" \
  --target-type instance \
  --health-check-protocol HTTP \
  --health-check-path "/" \
  --health-check-port traffic-port)
check_command "Failed to create target group"
TARGET_GROUP_ARN=$(echo "$TARGET_GROUP_RESULT" | jq -r
 '.TargetGroups[0].TargetGroupArn')
echo "Target group created with ARN: $TARGET_GROUP_ARN"

# Create load balancer
echo "Creating load balancer..."
LB_NAME="AppLoadBalancer-$RANDOM_ID"
LB_RESULT=$(aws elbv2 create-load-balancer \
  --name "$LB_NAME" \
```

```
  --subnets "$PUBLIC_SUBNET1_ID" "$PUBLIC_SUBNET2_ID" \
  --security-groups "$LB_SG_ID" \
  --tags "Key=Name,Value=$LB_NAME")
check_command "Failed to create load balancer"
LB_ARN=$(echo "$LB_RESULT" | jq -r '.LoadBalancers[0].LoadBalancerArn')
echo "Load balancer created with ARN: $LB_ARN"

# Wait for load balancer to be active
echo "Waiting for load balancer to be active..."
aws elbv2 wait load-balancer-available --load-balancer-arns "$LB_ARN"
check_command "Load balancer did not become available"

# Create listener
echo "Creating listener..."
LISTENER_RESULT=$(aws elbv2 create-listener \
  --load-balancer-arn "$LB_ARN" \
  --protocol HTTP \
  --port 80 \
  --default-actions "Type=forward,TargetGroupArn=$TARGET_GROUP_ARN")
check_command "Failed to create listener"
LISTENER_ARN=$(echo "$LISTENER_RESULT" | jq -r '.Listeners[0].ListenerArn')
echo "Listener created with ARN: $LISTENER_ARN"

# Create Auto Scaling group
echo "Creating Auto Scaling group..."
ASG_NAME="AppAutoScalingGroup-$RANDOM_ID"
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name "$ASG_NAME" \
  --launch-template "LaunchTemplateName=$LAUNCH_TEMPLATE_NAME,Version=\$Latest" \
  --min-size 2 \
  --max-size 4 \
  --desired-capacity 2 \
  --vpc-zone-identifier "$PRIVATE_SUBNET1_ID,$PRIVATE_SUBNET2_ID" \
  --target-group-arns "$TARGET_GROUP_ARN" \
  --health-check-type ELB \
  --health-check-grace-period 300 \
  --tags "Key=Name,Value=AppServer-$RANDOM_ID,PropagateAtLaunch=true"
check_command "Failed to create Auto Scaling group"
echo "Auto Scaling group created with name: $ASG_NAME"

# Get load balancer DNS name
LB_DNS_NAME=$(aws elbv2 describe-load-balancers --load-balancer-arns "$LB_ARN" --
query 'LoadBalancers[0].DNSName' --output text)
check_command "Failed to get load balancer DNS name"
```

```
echo ""
echo "==========================================="
echo "DEPLOYMENT COMPLETE"
echo "==========================================="
echo "VPC ID: $VPC_ID"
echo "Public Subnet 1: $PUBLIC_SUBNET1_ID (AZ: $AZ1)"
echo "Private Subnet 1: $PRIVATE_SUBNET1_ID (AZ: $AZ1)"
echo "Public Subnet 2: $PUBLIC_SUBNET2_ID (AZ: $AZ2)"
echo "Private Subnet 2: $PRIVATE_SUBNET2_ID (AZ: $AZ2)"
echo "NAT Gateway 1: $NAT_GW1_ID"
echo "NAT Gateway 2: $NAT_GW2_ID"
echo "Load Balancer: $LB_NAME"
echo "Auto Scaling Group: $ASG_NAME"
echo ""
echo "Your application will be available at: http://$LB_DNS_NAME"
echo "It may take a few minutes for the instances to launch and pass health checks."
echo ""

# Add health check monitoring
echo "==========================================="
echo "MONITORING INSTANCE HEALTH AND LOAD BALANCER"
echo "==========================================="
echo "Waiting for instances to launch and pass health checks..."
echo "This may take 3-5 minutes. Checking every 30 seconds..."

# Monitor instance health and load balancer accessibility
MAX_ATTEMPTS=10
ATTEMPT=1
HEALTHY_INSTANCES=0

while [ $ATTEMPT -le $MAX_ATTEMPTS ] && [ $HEALTHY_INSTANCES -lt 2 ]; do
  echo "Check attempt $ATTEMPT of $MAX_ATTEMPTS..."

  # Check Auto Scaling group instances
  echo "Checking Auto Scaling group instances..."
  ASG_INSTANCES=$(aws autoscaling describe-auto-scaling-groups --auto-
scaling-group-names "$ASG_NAME" --query 'AutoScalingGroups[0].Instances[*].
[InstanceId,HealthStatus]' --output json)
  echo "ASG Instances status:"
  echo "$ASG_INSTANCES" | jq -r '.[] | "Instance: \(.[0]), Health: \(.[1])"'

  # Check target group health
  echo "Checking target group health..."
```

```
  TARGET_HEALTH=$(aws elbv2 describe-target-health --target-group-arn
 "$TARGET_GROUP_ARN" --output json)
  echo "Target health status:"
  echo "$TARGET_HEALTH" | jq -r '.TargetHealthDescriptions[] | "Instance:
 \(.Target.Id), State: \(.TargetHealth.State), Reason: \(.TargetHealth.Reason // "N/
A"), Description: \(.TargetHealth.Description // "N/A")"'

  # Count healthy instances
  HEALTHY_INSTANCES=$(echo "$TARGET_HEALTH" | jq -r '[.TargetHealthDescriptions[] |
 select(.TargetHealth.State=="healthy")] | length')
  echo "Number of healthy instances: $HEALTHY_INSTANCES of 2 expected"

  # Check if we have healthy instances
  if [ $HEALTHY_INSTANCES -ge 2 ]; then
    echo "All instances are healthy!"

    # Test load balancer accessibility
    echo "Testing load balancer accessibility..."
    HTTP_STATUS=$(curl -s -o /dev/null -w "%{http_code}" "http://$LB_DNS_NAME")

    if [ "$HTTP_STATUS" = "200" ]; then
      echo "Load balancer is accessible! HTTP Status: $HTTP_STATUS"
      echo "You can access your application at: http://$LB_DNS_NAME"

      # Try to get the content to verify IMDSv2 is working
      echo "Fetching content to verify IMDSv2 functionality..."
      CONTENT=$(curl -s "http://$LB_DNS_NAME")
      echo "Response from server:"
      echo "$CONTENT"

      # Check if the content contains the expected pattern
      if [[ "$CONTENT" == *"Hello from"* && "$CONTENT" == *"in"* ]]; then
        echo "IMDSv2 is working correctly! The instance was able to access metadata
 using the token-based approach."
      else
        echo "Warning: Content doesn't match expected pattern. IMDSv2 functionality
 could not be verified."
      fi

      break
    else
      echo "Load balancer returned HTTP status: $HTTP_STATUS"
      echo "Will try again in 30 seconds..."
    fi
```

```
  else
    echo "Waiting for instances to become healthy..."
    echo "Will check again in 30 seconds..."
  fi

  ATTEMPT=$((ATTEMPT+1))

  if [ $ATTEMPT -le $MAX_ATTEMPTS ]; then
    sleep 30
  fi
done

if [ $HEALTHY_INSTANCES -lt 2 ]; then
  echo "Warning: Not all instances are healthy after maximum attempts."
  echo "You may need to wait longer or check for configuration issues."
fi

echo "To test your application, run:"
echo "curl http://$LB_DNS_NAME"
echo ""
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
  cleanup_resources
  echo "All resources have been deleted."
else
  echo "Resources will not be deleted. You can manually delete them later."
  echo "To delete resources, run this script again and choose to clean up."
fi
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AllocateAddress
  - AssociateRouteTable
  - AttachInternetGateway
  - AuthorizeSecurityGroupIngress

- CreateInternetGateway
- CreateLaunchTemplate
- CreateNatGateway
- CreateRoute
- CreateRouteTable
- CreateSecurityGroup
- CreateSubnet
- CreateVpc
- CreateVpcEndpoint
- DeleteAutoScalingGroup
- DeleteInternetGateway
- DeleteLaunchTemplate
- DeleteLoadBalancer
- DeleteNatGateway
- DeleteRouteTable
- DeleteSecurityGroup
- DeleteSubnet
- DeleteTargetGroup
- DeleteVpc
- DeleteVpcEndpoints
- DescribeAvailabilityZones
- DescribeImages
- DescribePrefixLists
- DetachInternetGateway
- ReleaseAddress

Commencer avec Amazon VPC

L'exemple de code suivant illustre comment :

- Configurez votre compte
- Création et configuration d'un VPC

- Configurer votre réseau
- Configuration de la sécurité
- Déployer les ressources
- Tester et vérifier
- Nettoyage des ressources
- Tenez compte des implications de production
- Tenez compte des implications de sécurité

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# VPC Creation Script
# This script creates a VPC with public and private subnets, internet gateway, NAT
 gateway, and security groups

# Set up logging
LOG_FILE="vpc_creation.log"
exec > >(tee -a "$LOG_FILE") 2>&1

# Function to handle errors
handle_error() {
  echo "ERROR: $1"
  echo "Resources created before error:"
  for resource in "${CREATED_RESOURCES[@]}"
  do
    echo "- $resource"
  done

  echo "Attempting to clean up resources..."
  cleanup_resources
  exit 1
```

```
}

# Function to clean up resources
cleanup_resources() {
  echo "Cleaning up resources in reverse order..."

  # Reverse the array to delete in reverse order of creation
  for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--))
  do
    resource="${CREATED_RESOURCES[$i]}"
    resource_type=$(echo "$resource" | cut -d':' -f1)
    resource_id=$(echo "$resource" | cut -d':' -f2)

    case "$resource_type" in
      "INSTANCE")
        echo "Terminating EC2 instance: $resource_id"
        aws ec2 terminate-instances --instance-ids "$resource_id" || echo "Failed to
 terminate instance: $resource_id"
        # Wait for instance to terminate
        echo "Waiting for instance to terminate..."
        aws ec2 wait instance-terminated --instance-ids "$resource_id" || echo
 "Failed to wait for instance termination: $resource_id"
        ;;
      "KEY_PAIR")
        echo "Deleting key pair: $resource_id"
        aws ec2 delete-key-pair --key-name "$resource_id" || echo "Failed to delete
 key pair: $resource_id"
        # Remove the .pem file if it exists
        if [ -f "${resource_id}.pem" ]; then
          rm -f "${resource_id}.pem"
        fi
        ;;
      "NAT_GATEWAY")
        echo "Deleting NAT Gateway: $resource_id"
        aws ec2 delete-nat-gateway --nat-gateway-id "$resource_id" || echo "Failed
 to delete NAT Gateway: $resource_id"
        # NAT Gateway deletion takes time, wait for it to complete
        echo "Waiting for NAT Gateway to be deleted..."
        aws ec2 wait nat-gateway-deleted --nat-gateway-ids "$resource_id" || echo
 "Failed to wait for NAT Gateway deletion: $resource_id"
        ;;
      "EIP")
        echo "Releasing Elastic IP: $resource_id"
```

```
        aws ec2 release-address --allocation-id "$resource_id" || echo "Failed to
 release Elastic IP: $resource_id"
        ;;
      "ROUTE_TABLE_ASSOCIATION")
        echo "Disassociating Route Table: $resource_id"
        aws ec2 disassociate-route-table --association-id "$resource_id" || echo
 "Failed to disassociate Route Table: $resource_id"
        ;;
      "ROUTE_TABLE")
        echo "Deleting Route Table: $resource_id"
        aws ec2 delete-route-table --route-table-id "$resource_id" || echo "Failed
 to delete Route Table: $resource_id"
        ;;
      "INTERNET_GATEWAY")
        echo "Detaching Internet Gateway: $resource_id from VPC: $VPC_ID"
        aws ec2 detach-internet-gateway --internet-gateway-id "$resource_id" --vpc-
id "$VPC_ID" || echo "Failed to detach Internet Gateway: $resource_id"
        echo "Deleting Internet Gateway: $resource_id"
        aws ec2 delete-internet-gateway --internet-gateway-id "$resource_id" || echo
 "Failed to delete Internet Gateway: $resource_id"
        ;;
      "SECURITY_GROUP")
        echo "Deleting Security Group: $resource_id"
        aws ec2 delete-security-group --group-id "$resource_id" || echo "Failed to
 delete Security Group: $resource_id"
        ;;
      "SUBNET")
        echo "Deleting Subnet: $resource_id"
        aws ec2 delete-subnet --subnet-id "$resource_id" || echo "Failed to delete
 Subnet: $resource_id"
        ;;
      "VPC")
        echo "Deleting VPC: $resource_id"
        aws ec2 delete-vpc --vpc-id "$resource_id" || echo "Failed to delete VPC:
 $resource_id"
        ;;
    esac
  done
}

# Initialize array to track created resources
CREATED_RESOURCES=()

echo "Starting VPC creation script at $(date)"
```

```
# Verify AWS CLI configuration
echo "Verifying AWS CLI configuration..."
aws configure list || handle_error "AWS CLI is not properly configured"

# Verify identity and permissions
echo "Verifying identity and permissions..."
if ! aws sts get-caller-identity; then
  echo "ERROR: Unable to verify AWS identity. This could be due to:"
  echo "  - Expired credentials"
  echo "  - Missing or invalid AWS credentials"
  echo "  - Insufficient permissions"
  echo ""
  echo "Please run 'aws configure' to update your credentials or check your IAM
 permissions."
  exit 1
fi

# Create VPC
echo "Creating VPC with CIDR block 10.0.0.0/16..."
VPC_ID=$(aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=MyVPC}]' --query 'Vpc.VpcId' --output text)

if [ -z "$VPC_ID" ]; then
  handle_error "Failed to create VPC"
fi

CREATED_RESOURCES+=("VPC:$VPC_ID")
echo "VPC created with ID: $VPC_ID"

# Enable DNS support and hostnames
echo "Enabling DNS support and hostnames for VPC..."
aws ec2 modify-vpc-attribute --vpc-id "$VPC_ID" --enable-dns-support || handle_error
 "Failed to enable DNS support"
aws ec2 modify-vpc-attribute --vpc-id "$VPC_ID" --enable-dns-hostnames ||
 handle_error "Failed to enable DNS hostnames"

# Get available Availability Zones
echo "Getting available Availability Zones..."
AZ1=$(aws ec2 describe-availability-zones --query 'AvailabilityZones[0].ZoneName' --
output text)
AZ2=$(aws ec2 describe-availability-zones --query 'AvailabilityZones[1].ZoneName' --
output text)
```

```
if [ -z "$AZ1" ] || [ -z "$AZ2" ]; then
  handle_error "Failed to get Availability Zones"
fi

echo "Using Availability Zones: $AZ1 and $AZ2"

# Create public subnets
echo "Creating public subnet in $AZ1..."
PUBLIC_SUBNET_AZ1=$(aws ec2 create-subnet \
  --vpc-id "$VPC_ID" \
  --cidr-block 10.0.0.0/24 \
  --availability-zone "$AZ1" \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Public-Subnet-
AZ1}]' \
  --query 'Subnet.SubnetId' \
  --output text)

if [ -z "$PUBLIC_SUBNET_AZ1" ]; then
  handle_error "Failed to create public subnet in AZ1"
fi

CREATED_RESOURCES+=("SUBNET:$PUBLIC_SUBNET_AZ1")
echo "Public subnet created in $AZ1 with ID: $PUBLIC_SUBNET_AZ1"

echo "Creating public subnet in $AZ2..."
PUBLIC_SUBNET_AZ2=$(aws ec2 create-subnet \
  --vpc-id "$VPC_ID" \
  --cidr-block 10.0.1.0/24 \
  --availability-zone "$AZ2" \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Public-Subnet-
AZ2}]' \
  --query 'Subnet.SubnetId' \
  --output text)

if [ -z "$PUBLIC_SUBNET_AZ2" ]; then
  handle_error "Failed to create public subnet in AZ2"
fi

CREATED_RESOURCES+=("SUBNET:$PUBLIC_SUBNET_AZ2")
echo "Public subnet created in $AZ2 with ID: $PUBLIC_SUBNET_AZ2"

# Create private subnets
echo "Creating private subnet in $AZ1..."
PRIVATE_SUBNET_AZ1=$(aws ec2 create-subnet \
```

```
  --vpc-id "$VPC_ID" \
  --cidr-block 10.0.2.0/24 \
  --availability-zone "$AZ1" \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Private-Subnet-
AZ1}]' \
  --query 'Subnet.SubnetId' \
  --output text)

if [ -z "$PRIVATE_SUBNET_AZ1" ]; then
  handle_error "Failed to create private subnet in AZ1"
fi

CREATED_RESOURCES+=("SUBNET:$PRIVATE_SUBNET_AZ1")
echo "Private subnet created in $AZ1 with ID: $PRIVATE_SUBNET_AZ1"

echo "Creating private subnet in $AZ2..."
PRIVATE_SUBNET_AZ2=$(aws ec2 create-subnet \
  --vpc-id "$VPC_ID" \
  --cidr-block 10.0.3.0/24 \
  --availability-zone "$AZ2" \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Private-Subnet-
AZ2}]' \
  --query 'Subnet.SubnetId' \
  --output text)

if [ -z "$PRIVATE_SUBNET_AZ2" ]; then
  handle_error "Failed to create private subnet in AZ2"
fi

CREATED_RESOURCES+=("SUBNET:$PRIVATE_SUBNET_AZ2")
echo "Private subnet created in $AZ2 with ID: $PRIVATE_SUBNET_AZ2"

# Create Internet Gateway
echo "Creating Internet Gateway..."
IGW_ID=$(aws ec2 create-internet-gateway \
  --tag-specifications 'ResourceType=internet-gateway,Tags=[{Key=Name,Value=MyIGW}]'
 \
  --query 'InternetGateway.InternetGatewayId' \
  --output text)

if [ -z "$IGW_ID" ]; then
  handle_error "Failed to create Internet Gateway"
fi
```

```
CREATED_RESOURCES+=("INTERNET_GATEWAY:$IGW_ID")
echo "Internet Gateway created with ID: $IGW_ID"

# Attach Internet Gateway to VPC
echo "Attaching Internet Gateway to VPC..."
aws ec2 attach-internet-gateway --internet-gateway-id "$IGW_ID" --vpc-id "$VPC_ID"
 || handle_error "Failed to attach Internet Gateway to VPC"

# Create public route table
echo "Creating public route table..."
PUBLIC_RT=$(aws ec2 create-route-table \
  --vpc-id "$VPC_ID" \
  --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Public-RT}]'
 \
  --query 'RouteTable.RouteTableId' \
  --output text)

if [ -z "$PUBLIC_RT" ]; then
  handle_error "Failed to create public route table"
fi

CREATED_RESOURCES+=("ROUTE_TABLE:$PUBLIC_RT")
echo "Public route table created with ID: $PUBLIC_RT"

# Add route to Internet Gateway
echo "Adding route to Internet Gateway in public route table..."
aws ec2 create-route --route-table-id "$PUBLIC_RT" --destination-cidr-block
 0.0.0.0/0 --gateway-id "$IGW_ID" || handle_error "Failed to add route to Internet
 Gateway"

# Associate public subnets with public route table
echo "Associating public subnet in $AZ1 with public route table..."
PUBLIC_RT_ASSOC_1=$(aws ec2 associate-route-table --route-table-id "$PUBLIC_RT" --
subnet-id "$PUBLIC_SUBNET_AZ1" --query 'AssociationId' --output text)

if [ -z "$PUBLIC_RT_ASSOC_1" ]; then
  handle_error "Failed to associate public subnet in AZ1 with public route table"
fi

CREATED_RESOURCES+=("ROUTE_TABLE_ASSOCIATION:$PUBLIC_RT_ASSOC_1")

echo "Associating public subnet in $AZ2 with public route table..."
PUBLIC_RT_ASSOC_2=$(aws ec2 associate-route-table --route-table-id "$PUBLIC_RT" --
subnet-id "$PUBLIC_SUBNET_AZ2" --query 'AssociationId' --output text)
```

```
if [ -z "$PUBLIC_RT_ASSOC_2" ]; then
  handle_error "Failed to associate public subnet in AZ2 with public route table"
fi

CREATED_RESOURCES+=("ROUTE_TABLE_ASSOCIATION:$PUBLIC_RT_ASSOC_2")

# Create private route table
echo "Creating private route table..."
PRIVATE_RT=$(aws ec2 create-route-table \
  --vpc-id "$VPC_ID" \
  --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Private-RT}]'
 \
  --query 'RouteTable.RouteTableId' \
  --output text)

if [ -z "$PRIVATE_RT" ]; then
  handle_error "Failed to create private route table"
fi

CREATED_RESOURCES+=("ROUTE_TABLE:$PRIVATE_RT")
echo "Private route table created with ID: $PRIVATE_RT"

# Associate private subnets with private route table
echo "Associating private subnet in $AZ1 with private route table..."
PRIVATE_RT_ASSOC_1=$(aws ec2 associate-route-table --route-table-id "$PRIVATE_RT" --
subnet-id "$PRIVATE_SUBNET_AZ1" --query 'AssociationId' --output text)

if [ -z "$PRIVATE_RT_ASSOC_1" ]; then
  handle_error "Failed to associate private subnet in AZ1 with private route table"
fi

CREATED_RESOURCES+=("ROUTE_TABLE_ASSOCIATION:$PRIVATE_RT_ASSOC_1")

echo "Associating private subnet in $AZ2 with private route table..."
PRIVATE_RT_ASSOC_2=$(aws ec2 associate-route-table --route-table-id "$PRIVATE_RT" --
subnet-id "$PRIVATE_SUBNET_AZ2" --query 'AssociationId' --output text)

if [ -z "$PRIVATE_RT_ASSOC_2" ]; then
  handle_error "Failed to associate private subnet in AZ2 with private route table"
fi

CREATED_RESOURCES+=("ROUTE_TABLE_ASSOCIATION:$PRIVATE_RT_ASSOC_2")
```

```
# Allocate Elastic IP for NAT Gateway
echo "Allocating Elastic IP for NAT Gateway..."
EIP_ALLOC=$(aws ec2 allocate-address --domain vpc --query 'AllocationId' --output
 text)

if [ -z "$EIP_ALLOC" ]; then
  handle_error "Failed to allocate Elastic IP"
fi

CREATED_RESOURCES+=("EIP:$EIP_ALLOC")
echo "Elastic IP allocated with ID: $EIP_ALLOC"

# Create NAT Gateway
echo "Creating NAT Gateway in public subnet in $AZ1..."
NAT_GW=$(aws ec2 create-nat-gateway \
  --subnet-id "$PUBLIC_SUBNET_AZ1" \
  --allocation-id "$EIP_ALLOC" \
  --tag-specifications
 'ResourceType=natgateway,Tags=[{Key=Name,Value=MyNATGateway}]' \
  --query 'NatGateway.NatGatewayId' \
  --output text)

if [ -z "$NAT_GW" ]; then
  handle_error "Failed to create NAT Gateway"
fi

CREATED_RESOURCES+=("NAT_GATEWAY:$NAT_GW")
echo "NAT Gateway created with ID: $NAT_GW"

# Wait for NAT Gateway to be available
echo "Waiting for NAT Gateway to be available..."
aws ec2 wait nat-gateway-available --nat-gateway-ids "$NAT_GW" || handle_error "NAT
 Gateway did not become available"

# Add route to NAT Gateway in private route table
echo "Adding route to NAT Gateway in private route table..."
aws ec2 create-route --route-table-id "$PRIVATE_RT" --destination-cidr-block
 0.0.0.0/0 --nat-gateway-id "$NAT_GW" || handle_error "Failed to add route to NAT
 Gateway"

# Enable auto-assign public IP for instances in public subnets
echo "Enabling auto-assign public IP for instances in public subnet in $AZ1..."
```

```
aws ec2 modify-subnet-attribute --subnet-id "$PUBLIC_SUBNET_AZ1" --map-public-ip-on-
launch || handle_error "Failed to enable auto-assign public IP for public subnet in
 AZ1"

echo "Enabling auto-assign public IP for instances in public subnet in $AZ2..."
aws ec2 modify-subnet-attribute --subnet-id "$PUBLIC_SUBNET_AZ2" --map-public-ip-on-
launch || handle_error "Failed to enable auto-assign public IP for public subnet in
 AZ2"

# Create security group for web servers
echo "Creating security group for web servers..."
WEB_SG=$(aws ec2 create-security-group \
  --group-name "WebServerSG-$(date +%s)" \
  --description "Security group for web servers" \
  --vpc-id "$VPC_ID" \
  --query 'GroupId' \
  --output text)

if [ -z "$WEB_SG" ]; then
  handle_error "Failed to create security group for web servers"
fi

CREATED_RESOURCES+=("SECURITY_GROUP:$WEB_SG")
echo "Security group for web servers created with ID: $WEB_SG"

# Allow HTTP and HTTPS traffic
echo "Allowing HTTP traffic to web servers security group..."
aws ec2 authorize-security-group-ingress --group-id "$WEB_SG" --protocol tcp --port
 80 --cidr 0.0.0.0/0 || handle_error "Failed to allow HTTP traffic"

echo "Allowing HTTPS traffic to web servers security group..."
aws ec2 authorize-security-group-ingress --group-id "$WEB_SG" --protocol tcp --port
 443 --cidr 0.0.0.0/0 || handle_error "Failed to allow HTTPS traffic"

# Note: In a production environment, you should restrict the source IP ranges for
 security
echo "NOTE: In a production environment, you should restrict the source IP ranges
 for HTTP and HTTPS traffic"

# Create security group for database servers
echo "Creating security group for database servers..."
DB_SG=$(aws ec2 create-security-group \
  --group-name "DBServerSG-$(date +%s)" \
  --description "Security group for database servers" \
```

```
  --vpc-id "$VPC_ID" \
  --query 'GroupId' \
  --output text)

if [ -z "$DB_SG" ]; then
  handle_error "Failed to create security group for database servers"
fi

CREATED_RESOURCES+=("SECURITY_GROUP:$DB_SG")
echo "Security group for database servers created with ID: $DB_SG"

# Allow MySQL/Aurora traffic from web servers only
echo "Allowing MySQL/Aurora traffic from web servers to database servers..."
aws ec2 authorize-security-group-ingress --group-id "$DB_SG" --protocol tcp --
port 3306 --source-group "$WEB_SG" || handle_error "Failed to allow MySQL/Aurora
 traffic"

# Verify VPC configuration
echo "Verifying VPC configuration..."
echo "VPC:"
aws ec2 describe-vpcs --vpc-id "$VPC_ID" || handle_error "Failed to describe VPC"

echo "Subnets:"
aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID" || handle_error
 "Failed to describe subnets"

echo "Route tables:"
aws ec2 describe-route-tables --filters "Name=vpc-id,Values=$VPC_ID" || handle_error
 "Failed to describe route tables"

echo "Internet gateway:"
aws ec2 describe-internet-gateways --filters "Name=attachment.vpc-id,Values=$VPC_ID"
 || handle_error "Failed to describe Internet Gateway"

echo "NAT gateway:"
aws ec2 describe-nat-gateways --filter "Name=vpc-id,Values=$VPC_ID" || handle_error
 "Failed to describe NAT Gateway"

echo "Security groups:"
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=$VPC_ID" ||
 handle_error "Failed to describe security groups"

echo ""
# Summary of created resources
```

```
echo "VPC creation completed successfully!"
echo "Summary of created resources:"
echo "- VPC: $VPC_ID"
echo "- Public Subnet in $AZ1: $PUBLIC_SUBNET_AZ1"
echo "- Public Subnet in $AZ2: $PUBLIC_SUBNET_AZ2"
echo "- Private Subnet in $AZ1: $PRIVATE_SUBNET_AZ1"
echo "- Private Subnet in $AZ2: $PRIVATE_SUBNET_AZ2"
echo "- Internet Gateway: $IGW_ID"
echo "- Public Route Table: $PUBLIC_RT"
echo "- Private Route Table: $PRIVATE_RT"
echo "- Elastic IP: $EIP_ALLOC"
echo "- NAT Gateway: $NAT_GW"
echo "- Web Servers Security Group: $WEB_SG"
echo "- Database Servers Security Group: $DB_SG"

# Deploy EC2 instances
echo ""
echo "Deploying EC2 instances..."

# Create key pair for SSH access
KEY_NAME="vpc-tutorial-key-$(date +%s)"
echo "Creating key pair $KEY_NAME..."
aws ec2 create-key-pair --key-name "$KEY_NAME" --query 'KeyMaterial' --output text >
 "${KEY_NAME}.pem" || handle_error "Failed to create key pair"
chmod 400 "${KEY_NAME}.pem"
echo "Key pair saved to ${KEY_NAME}.pem"
CREATED_RESOURCES+=("KEY_PAIR:$KEY_NAME")

# Get latest Amazon Linux 2 AMI
echo "Getting latest Amazon Linux 2 AMI..."
AMI_ID=$(aws ec2 describe-images --owners amazon \
  --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2"
 "Name=state,Values=available" \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" --output text) ||
 handle_error "Failed to get AMI"
echo "Using AMI: $AMI_ID"

# Launch web server in public subnet
echo "Launching web server in public subnet..."
WEB_INSTANCE=$(aws ec2 run-instances \
  --image-id "$AMI_ID" \
  --count 1 \
  --instance-type t2.micro \
  --key-name "$KEY_NAME" \
```

```
  --security-group-ids "$WEB_SG" \
  --subnet-id "$PUBLIC_SUBNET_AZ1" \
  --associate-public-ip-address \
  --user-data '#!/bin/bash
    yum update -y
    yum install -y httpd
    systemctl start httpd
    systemctl enable httpd
    echo "<h1>Hello from $(hostname -f) in the public subnet</h1>" > /var/www/html/
index.html' \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WebServer}]' \
  --query 'Instances[0].InstanceId' \
  --output text) || handle_error "Failed to launch web server"
echo "Web server instance created with ID: $WEB_INSTANCE"
CREATED_RESOURCES+=("INSTANCE:$WEB_INSTANCE")

# Wait for web server to be running
echo "Waiting for web server to be running..."
aws ec2 wait instance-running --instance-ids "$WEB_INSTANCE"

# Get web server public IP
WEB_PUBLIC_IP=$(aws ec2 describe-instances --instance-ids "$WEB_INSTANCE" \
  --query 'Reservations[0].Instances[0].PublicIpAddress' --output text)
echo "Web server public IP: $WEB_PUBLIC_IP"
echo "You can access the web server at: http://$WEB_PUBLIC_IP"

# Launch database server in private subnet
echo "Launching database server in private subnet..."
DB_INSTANCE=$(aws ec2 run-instances \
  --image-id "$AMI_ID" \
  --count 1 \
  --instance-type t2.micro \
  --key-name "$KEY_NAME" \
  --security-group-ids "$DB_SG" \
  --subnet-id "$PRIVATE_SUBNET_AZ1" \
  --user-data '#!/bin/bash
    yum update -y
    yum install -y mariadb-server
    systemctl start mariadb
    systemctl enable mariadb' \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=DBServer}]' \
  --query 'Instances[0].InstanceId' \
  --output text) || handle_error "Failed to launch database server"
echo "Database server instance created with ID: $DB_INSTANCE"
```

```
CREATED_RESOURCES+=("INSTANCE:$DB_INSTANCE")

# Wait for database server to be running
echo "Waiting for database server to be running..."
aws ec2 wait instance-running --instance-ids "$DB_INSTANCE"

# Get database server private IP
DB_PRIVATE_IP=$(aws ec2 describe-instances --instance-ids "$DB_INSTANCE" \
  --query 'Reservations[0].Instances[0].PrivateIpAddress' --output text)
echo "Database server private IP: $DB_PRIVATE_IP"

echo "EC2 instances deployed successfully!"
echo "- Web Server (Public): $WEB_INSTANCE ($WEB_PUBLIC_IP)"
echo "- Database Server (Private): $DB_INSTANCE ($DB_PRIVATE_IP)"
echo ""
echo "Note: To connect to the web server: ssh -i ${KEY_NAME}.pem ec2-user@
$WEB_PUBLIC_IP"
echo "To connect to the database server, you must first connect to the web server,
 then use it as a bastion host."
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE
if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
  echo "Cleaning up resources..."
  cleanup_resources
  echo "All resources have been cleaned up."
else
  echo "Resources will not be cleaned up. You can manually clean them up later."
fi

echo "Script completed at $(date)"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AllocateAddress
  - AssociateRouteTable
  - AttachInternetGateway
  - AuthorizeSecurityGroupIngress

- CreateInternetGateway
- CreateKeyPair
- CreateNatGateway
- CreateRoute
- CreateRouteTable
- CreateSecurityGroup
- CreateSubnet
- CreateVpc
- DeleteInternetGateway
- DeleteKeyPair
- DeleteNatGateway
- DeleteRouteTable
- DeleteSecurityGroup
- DeleteSubnet
- DeleteVpc
- DescribeAvailabilityZones
- DescribeImages
- DescribeInstances
- DescribeInternetGateways
- DescribeNatGateways
- DescribeRouteTables
- DescribeSecurityGroups
- DescribeSubnets
- DescribeVpcs
- DetachInternetGateway
- DisassociateRouteTable
- ModifySubnetAttribute
- ModifyVpcAttribute
- ReleaseAddress
- RunInstances

- TerminateInstances

Commencez avec Transit Gateway

L'exemple de code suivant illustre comment :

- Créez une passerelle de transit avec support DNS et paramètres de table de routage par défaut
- Attendez que la passerelle de transit soit disponible
- Attachez-en deux VPCs à la passerelle de transit à l'aide de sous-réseaux
- Attendez que les pièces jointes VPC soient disponibles
- Ajouter des itinéraires entre deux VPCs points via la passerelle de transit
- Tester la connectivité entre les ressources VPC
- Nettoyez les ressources, y compris les itinéraires, les pièces jointes et les passerelles de transit

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# Amazon VPC Transit Gateway CLI Script
# This script demonstrates how to create a transit gateway and connect two VPCs
# Modified to work with older AWS CLI versions that don't support transit gateway
 wait commands

# Error handling
set -e
LOG_FILE="transit-gateway-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

# Function to wait for transit gateway to be available
wait_for_tgw() {
  local tgw_id=$1
  echo "Waiting for Transit Gateway $tgw_id to become available..."
```

```
  while true; do
    status=$(aws ec2 describe-transit-gateways --transit-gateway-ids "$tgw_id" --
query "TransitGateways[0].State" --output text)
    echo "Current status: $status"

    if [ "$status" = "available" ]; then
      echo "Transit Gateway is now available"
      break
    fi

    echo "Waiting for transit gateway to become available. Current state: $status"
    sleep 10
  done
}

# Function to wait for transit gateway attachment to be available
wait_for_tgw_attachment() {
  local attachment_id=$1
  echo "Waiting for Transit Gateway Attachment $attachment_id to become
 available..."

  while true; do
    status=$(aws ec2 describe-transit-gateway-vpc-attachments --transit-gateway-
attachment-ids "$attachment_id" --query "TransitGatewayVpcAttachments[0].State" --
output text)
    echo "Current status: $status"

    if [ "$status" = "available" ]; then
      echo "Transit Gateway Attachment is now available"
      break
    fi

    echo "Waiting for transit gateway attachment to become available. Current state:
 $status"
    sleep 10
  done
}

# Function to wait for transit gateway attachment to be deleted
wait_for_tgw_attachment_deleted() {
  local attachment_id=$1
  echo "Waiting for Transit Gateway Attachment $attachment_id to be deleted..."
```

```
  while true; do
    # Check if the attachment still exists
    count=$(aws ec2 describe-transit-gateway-vpc-attachments --filters
 "Name=transit-gateway-attachment-id,Values=$attachment_id" --query
 "length(TransitGatewayVpcAttachments)" --output text)

    if [ "$count" = "0" ]; then
      echo "Transit Gateway Attachment has been deleted"
      break
    fi

    status=$(aws ec2 describe-transit-gateway-vpc-attachments --transit-gateway-
attachment-ids "$attachment_id" --query "TransitGatewayVpcAttachments[0].State" --
output text 2>/dev/null || echo "deleted")

    if [ "$status" = "deleted" ]; then
      echo "Transit Gateway Attachment has been deleted"
      break
    fi

    echo "Waiting for transit gateway attachment to be deleted. Current state:
 $status"
    sleep 10
  done
}

# Function to clean up resources
cleanup() {
  echo "Error occurred. Cleaning up resources..."

  # Delete resources in reverse order
  if [ ! -z "$TGW_ATTACHMENT_1_ID" ]; then
    echo "Deleting Transit Gateway VPC Attachment 1: $TGW_ATTACHMENT_1_ID"
    aws ec2 delete-transit-gateway-vpc-attachment --transit-gateway-attachment-id
 "$TGW_ATTACHMENT_1_ID" || true
    wait_for_tgw_attachment_deleted "$TGW_ATTACHMENT_1_ID" || true
  fi

  if [ ! -z "$TGW_ATTACHMENT_2_ID" ]; then
    echo "Deleting Transit Gateway VPC Attachment 2: $TGW_ATTACHMENT_2_ID"
    aws ec2 delete-transit-gateway-vpc-attachment --transit-gateway-attachment-id
 "$TGW_ATTACHMENT_2_ID" || true
    wait_for_tgw_attachment_deleted "$TGW_ATTACHMENT_2_ID" || true
  fi
```

```
  if [ ! -z "$TGW_ID" ]; then
    echo "Deleting Transit Gateway: $TGW_ID"
    aws ec2 delete-transit-gateway --transit-gateway-id "$TGW_ID" || true
  fi

  exit 1
}

# Set up trap for error handling
trap cleanup ERR

echo "=== Amazon VPC Transit Gateway Tutorial ==="
echo "This script will create a transit gateway and connect two VPCs"
echo ""

# Get a valid availability zone dynamically
echo "Getting available AZ in current region..."
AZ=$(aws ec2 describe-availability-zones --query "AvailabilityZones[0].ZoneName" --
output text)
echo "Using availability zone: $AZ"

# Check if VPCs exist
echo "Checking for existing VPCs..."
VPC1_ID=$(aws ec2 describe-vpcs --filters "Name=tag:Name,Values=VPC1" --query
 "Vpcs[0].VpcId" --output text)
VPC2_ID=$(aws ec2 describe-vpcs --filters "Name=tag:Name,Values=VPC2" --query
 "Vpcs[0].VpcId" --output text)

if [ "$VPC1_ID" == "None" ] || [ -z "$VPC1_ID" ]; then
  echo "Creating VPC1..."
  VPC1_ID=$(aws ec2 create-vpc --cidr-block 10.1.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=VPC1}]' --query Vpc.VpcId --output text)
  echo "Created VPC1: $VPC1_ID"

  # Create a subnet in VPC1
  echo "Creating subnet in VPC1..."
  SUBNET1_ID=$(aws ec2 create-subnet --vpc-id "$VPC1_ID" --cidr-
block 10.1.0.0/24 --availability-zone "$AZ" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=VPC1-Subnet}]' --query Subnet.SubnetId
 --output text)
  echo "Created subnet in VPC1: $SUBNET1_ID"
else
  echo "Using existing VPC1: $VPC1_ID"
```

```
  SUBNET1_ID=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC1_ID" --
query "Subnets[0].SubnetId" --output text)
  if [ "$SUBNET1_ID" == "None" ] || [ -z "$SUBNET1_ID" ]; then
    echo "Creating subnet in VPC1..."
    SUBNET1_ID=$(aws ec2 create-subnet --vpc-id "$VPC1_ID" --cidr-
block 10.1.0.0/24 --availability-zone "$AZ" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=VPC1-Subnet}]' --query Subnet.SubnetId
 --output text)
    echo "Created subnet in VPC1: $SUBNET1_ID"
  else
    echo "Using existing subnet in VPC1: $SUBNET1_ID"
  fi
fi

if [ "$VPC2_ID" == "None" ] || [ -z "$VPC2_ID" ]; then
  echo "Creating VPC2..."
  VPC2_ID=$(aws ec2 create-vpc --cidr-block 10.2.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=VPC2}]' --query Vpc.VpcId --output text)
  echo "Created VPC2: $VPC2_ID"

  # Create a subnet in VPC2
  echo "Creating subnet in VPC2..."
  SUBNET2_ID=$(aws ec2 create-subnet --vpc-id "$VPC2_ID" --cidr-
block 10.2.0.0/24 --availability-zone "$AZ" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=VPC2-Subnet}]' --query Subnet.SubnetId
 --output text)
  echo "Created subnet in VPC2: $SUBNET2_ID"
else
  echo "Using existing VPC2: $VPC2_ID"
  SUBNET2_ID=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC2_ID" --
query "Subnets[0].SubnetId" --output text)
  if [ "$SUBNET2_ID" == "None" ] || [ -z "$SUBNET2_ID" ]; then
    echo "Creating subnet in VPC2..."
    SUBNET2_ID=$(aws ec2 create-subnet --vpc-id "$VPC2_ID" --cidr-
block 10.2.0.0/24 --availability-zone "$AZ" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=VPC2-Subnet}]' --query Subnet.SubnetId
 --output text)
    echo "Created subnet in VPC2: $SUBNET2_ID"
  else
    echo "Using existing subnet in VPC2: $SUBNET2_ID"
  fi
fi

# Get route tables for each VPC
```

```
RTB1_ID=$(aws ec2 describe-route-tables --filters "Name=vpc-id,Values=$VPC1_ID" --
query "RouteTables[0].RouteTableId" --output text)
RTB2_ID=$(aws ec2 describe-route-tables --filters "Name=vpc-id,Values=$VPC2_ID" --
query "RouteTables[0].RouteTableId" --output text)

echo "Route table for VPC1: $RTB1_ID"
echo "Route table for VPC2: $RTB2_ID"

# Step 1: Create the transit gateway
echo "Creating Transit Gateway..."
TGW_ID=$(aws ec2 create-transit-gateway \
  --description "My Transit Gateway" \
  --options
 AmazonSideAsn=64512,AutoAcceptSharedAttachments=disable,DefaultRouteTableAssociation=enable
 \
  --tag-specifications 'ResourceType=transit-
gateway,Tags=[{Key=Name,Value=MyTransitGateway}]' \
  --query TransitGateway.TransitGatewayId \
  --output text)

echo "Created Transit Gateway: $TGW_ID"

# Wait for the transit gateway to become available
wait_for_tgw "$TGW_ID"

# Step 2: Attach VPCs to the transit gateway
echo "Attaching VPC1 to Transit Gateway..."
TGW_ATTACHMENT_1_ID=$(aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id "$TGW_ID" \
  --vpc-id "$VPC1_ID" \
  --subnet-ids "$SUBNET1_ID" \
  --tag-specifications 'ResourceType=transit-gateway-
attachment,Tags=[{Key=Name,Value=VPC1-Attachment}]' \
  --query TransitGatewayVpcAttachment.TransitGatewayAttachmentId \
  --output text)

echo "Created Transit Gateway VPC Attachment for VPC1: $TGW_ATTACHMENT_1_ID"

echo "Attaching VPC2 to Transit Gateway..."
TGW_ATTACHMENT_2_ID=$(aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id "$TGW_ID" \
  --vpc-id "$VPC2_ID" \
  --subnet-ids "$SUBNET2_ID" \
```

```
  --tag-specifications 'ResourceType=transit-gateway-
attachment,Tags=[{Key=Name,Value=VPC2-Attachment}]' \
  --query TransitGatewayVpcAttachment.TransitGatewayAttachmentId \
  --output text)

echo "Created Transit Gateway VPC Attachment for VPC2: $TGW_ATTACHMENT_2_ID"

# Wait for the attachments to become available
wait_for_tgw_attachment "$TGW_ATTACHMENT_1_ID"
wait_for_tgw_attachment "$TGW_ATTACHMENT_2_ID"

# Step 3: Add routes between the transit gateway and VPCs
echo "Adding route from VPC1 to VPC2 via Transit Gateway..."
aws ec2 create-route \
  --route-table-id "$RTB1_ID" \
  --destination-cidr-block 10.2.0.0/16 \
  --transit-gateway-id "$TGW_ID"

echo "Adding route from VPC2 to VPC1 via Transit Gateway..."
aws ec2 create-route \
  --route-table-id "$RTB2_ID" \
  --destination-cidr-block 10.1.0.0/16 \
  --transit-gateway-id "$TGW_ID"

echo "Routes added successfully"

# Step 4: Display information for testing
echo ""
echo "=== Transit Gateway Setup Complete ==="
echo "Transit Gateway ID: $TGW_ID"
echo "VPC1 ID: $VPC1_ID"
echo "VPC2 ID: $VPC2_ID"
echo ""
echo "To test connectivity:"
echo "1. Launch an EC2 instance in each VPC"
echo "2. Configure security groups to allow ICMP traffic"
echo "3. Connect to one instance and ping the other instance's private IP"
echo ""

# Prompt user before cleanup
read -p "Press Enter to view created resources, or Ctrl+C to exit without
 cleanup..."

echo ""
```

```
echo "=== Resources Created ==="
echo "Transit Gateway: $TGW_ID"
echo "VPC1: $VPC1_ID"
echo "VPC2: $VPC2_ID"
echo "Subnet in VPC1: $SUBNET1_ID"
echo "Subnet in VPC2: $SUBNET2_ID"
echo "Transit Gateway Attachment for VPC1: $TGW_ATTACHMENT_1_ID"
echo "Transit Gateway Attachment for VPC2: $TGW_ATTACHMENT_2_ID"
echo ""

read -p "Do you want to clean up these resources? (y/n): " CLEANUP_CONFIRM
if [[ $CLEANUP_CONFIRM == "y" || $CLEANUP_CONFIRM == "Y" ]]; then
  echo "Starting cleanup..."

  # Delete routes
  echo "Deleting routes..."
  aws ec2 delete-route --route-table-id "$RTB1_ID" --destination-cidr-block
 10.2.0.0/16
  aws ec2 delete-route --route-table-id "$RTB2_ID" --destination-cidr-block
 10.1.0.0/16

  # Delete transit gateway attachments
  echo "Deleting Transit Gateway VPC Attachment for VPC1: $TGW_ATTACHMENT_1_ID"
  aws ec2 delete-transit-gateway-vpc-attachment --transit-gateway-attachment-id
 "$TGW_ATTACHMENT_1_ID"

  echo "Deleting Transit Gateway VPC Attachment for VPC2: $TGW_ATTACHMENT_2_ID"
  aws ec2 delete-transit-gateway-vpc-attachment --transit-gateway-attachment-id
 "$TGW_ATTACHMENT_2_ID"

  # Wait for attachments to be deleted
  wait_for_tgw_attachment_deleted "$TGW_ATTACHMENT_1_ID"
  wait_for_tgw_attachment_deleted "$TGW_ATTACHMENT_2_ID"

  # Delete transit gateway
  echo "Deleting Transit Gateway: $TGW_ID"
  aws ec2 delete-transit-gateway --transit-gateway-id "$TGW_ID"

  echo "Cleanup completed successfully"
else
  echo "Skipping cleanup. Resources will continue to incur charges until manually
 deleted."
fi
```

```
echo "Tutorial completed. See $LOG_FILE for detailed logs."
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateRoute
  - CreateSubnet
  - CreateTransitGateway
  - CreateTransitGatewayVpcAttachment
  - CreateVpc
  - DeleteRoute
  - DeleteTransitGateway
  - DeleteTransitGatewayVpcAttachment
  - DescribeAvailabilityZones
  - DescribeRouteTables
  - DescribeSubnets
  - DescribeTransitGatewayAttachments
  - DescribeTransitGatewayVpcAttachments
  - DescribeTransitGateways
  - DescribeVpcs

Commencez avec VPC IPAM

L'exemple de code suivant illustre comment :

- Configurez et configurez le gestionnaire d'adresses IP Amazon VPC (IPAM) à l'aide de la CLI.
- Créez un IPAM avec des régions d'exploitation (par exemple, us-east-1, us-west-2).
- Récupérez l'ID de portée privée pour l'IPAM.
- Créez une structure hiérarchique de IPv4 pools (pools de haut niveau, régionaux et de développement).
- Fournissez des blocs CIDR à chaque pool (par exemple, 10.0.0.0/8, 10.0.0.0/16, 10.0.0.0/24).
- Créez un VPC à l'aide d'un CIDR alloué à partir d'un pool IPAM.
- Vérifiez les allocations de pool IPAM et la création de VPC.

- Résolvez les problèmes courants tels que les erreurs d'autorisation, les échecs d'allocation CIDR et les violations de dépendance.
- Nettoyez les ressources IPAM (VPC CIDRs, pools et IPAM) pour éviter des frais inutiles.
- Découvrez les prochaines étapes pour découvrir les fonctionnalités IPAM avancées.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# IPAM Getting Started CLI Script - Version 7
# This script creates an IPAM, creates a hierarchy of IP address pools, and
 allocates a CIDR to a VPC
# Fixed to correctly identify the private scope ID, wait for resources to be
 available, add locale to development pool,
# use the correct parameter names for VPC creation, and wait for CIDR provisioning
 to complete

# Set up logging
LOG_FILE="ipam_script.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting IPAM setup script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Function to clean up resources
cleanup_resources() {
```

```
    echo ""
    echo "========================================="
    echo "RESOURCES CREATED:"
    echo "========================================="

    if [ -n "$VPC_ID" ]; then
        echo "VPC: $VPC_ID"
    fi

    if [ -n "$DEV_POOL_ID" ]; then
        echo "Development Pool: $DEV_POOL_ID"
    fi

    if [ -n "$REGIONAL_POOL_ID" ]; then
        echo "Regional Pool: $REGIONAL_POOL_ID"
    fi

    if [ -n "$TOP_POOL_ID" ]; then
        echo "Top-level Pool: $TOP_POOL_ID"
    fi

    if [ -n "$IPAM_ID" ]; then
        echo "IPAM: $IPAM_ID"
    fi

    echo ""
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    echo "Do you want to clean up all created resources? (y/n): "
    read -r CLEANUP_CHOICE

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        echo "Starting cleanup..."

        # Delete resources in reverse order of creation to handle dependencies

        if [ -n "$VPC_ID" ]; then
            echo "Deleting VPC: $VPC_ID"
            aws ec2 delete-vpc --vpc-id "$VPC_ID" || echo "Failed to delete VPC"
            echo "Waiting for VPC to be deleted..."
            sleep 10
        fi
```

```
        if [ -n "$DEV_POOL_ID" ]; then
            echo "Deleting Development Pool: $DEV_POOL_ID"
            # First deprovision any CIDRs from the pool
            CIDRS=$(aws ec2 get-ipam-pool-cidrs --ipam-pool-id "$DEV_POOL_ID" --
query 'IpamPoolCidrs[].Cidr' --output text)
            for CIDR in $CIDRS; do
                echo "Deprovisioning CIDR $CIDR from Development Pool"
                aws ec2 deprovision-ipam-pool-cidr --ipam-pool-id "$DEV_POOL_ID" --
cidr "$CIDR" || echo "Failed to deprovision CIDR $CIDR"
                sleep 5
            done
            aws ec2 delete-ipam-pool --ipam-pool-id "$DEV_POOL_ID" || echo "Failed
 to delete Development Pool"
            echo "Waiting for Development Pool to be deleted..."
            sleep 10
        fi

        if [ -n "$REGIONAL_POOL_ID" ]; then
            echo "Deleting Regional Pool: $REGIONAL_POOL_ID"
            # First deprovision any CIDRs from the pool
            CIDRS=$(aws ec2 get-ipam-pool-cidrs --ipam-pool-id "$REGIONAL_POOL_ID"
 --query 'IpamPoolCidrs[].Cidr' --output text)
            for CIDR in $CIDRS; do
                echo "Deprovisioning CIDR $CIDR from Regional Pool"
                aws ec2 deprovision-ipam-pool-cidr --ipam-pool-id
 "$REGIONAL_POOL_ID" --cidr "$CIDR" || echo "Failed to deprovision CIDR $CIDR"
                sleep 5
            done
            aws ec2 delete-ipam-pool --ipam-pool-id "$REGIONAL_POOL_ID" || echo
 "Failed to delete Regional Pool"
            echo "Waiting for Regional Pool to be deleted..."
            sleep 10
        fi

        if [ -n "$TOP_POOL_ID" ]; then
            echo "Deleting Top-level Pool: $TOP_POOL_ID"
            # First deprovision any CIDRs from the pool
            CIDRS=$(aws ec2 get-ipam-pool-cidrs --ipam-pool-id "$TOP_POOL_ID" --
query 'IpamPoolCidrs[].Cidr' --output text)
            for CIDR in $CIDRS; do
                echo "Deprovisioning CIDR $CIDR from Top-level Pool"
                aws ec2 deprovision-ipam-pool-cidr --ipam-pool-id "$TOP_POOL_ID" --
cidr "$CIDR" || echo "Failed to deprovision CIDR $CIDR"
                sleep 5
```

```
            done
            aws ec2 delete-ipam-pool --ipam-pool-id "$TOP_POOL_ID" || echo "Failed
 to delete Top-level Pool"
            echo "Waiting for Top-level Pool to be deleted..."
            sleep 10
        fi

        if [ -n "$IPAM_ID" ]; then
            echo "Deleting IPAM: $IPAM_ID"
            aws ec2 delete-ipam --ipam-id "$IPAM_ID" || echo "Failed to delete IPAM"
        fi

        echo "Cleanup completed."
    else
        echo "Cleanup skipped. Resources will remain in your account."
    fi
}

# Function to wait for a pool to be in the 'create-complete' state
wait_for_pool() {
    local pool_id=$1
    local max_attempts=30
    local attempt=1
    local state=""

    echo "Waiting for pool $pool_id to be available..."

    while [ $attempt -le $max_attempts ]; do
        state=$(aws ec2 describe-ipam-pools --ipam-pool-ids "$pool_id" --query
 'IpamPools[0].State' --output text)

        if [ "$state" = "create-complete" ]; then
            echo "Pool $pool_id is now available (state: $state)"
            return 0
        fi

        echo "Attempt $attempt/$max_attempts: Pool $pool_id is in state: $state.
 Waiting..."
        sleep 10
        ((attempt++))
    done

    echo "Timed out waiting for pool $pool_id to be available"
    return 1
```

```
}

# Function to wait for a CIDR to be fully provisioned
wait_for_cidr_provisioning() {
    local pool_id=$1
    local cidr=$2
    local max_attempts=30
    local attempt=1
    local state=""

    echo "Waiting for CIDR $cidr to be fully provisioned in pool $pool_id..."

    while [ $attempt -le $max_attempts ]; do
        state=$(aws ec2 get-ipam-pool-cidrs --ipam-pool-id "$pool_id" --query
 "IpamPoolCidrs[?Cidr=='$cidr'].State" --output text)

        if [ "$state" = "provisioned" ]; then
            echo "CIDR $cidr is now fully provisioned (state: $state)"
            return 0
        fi

        echo "Attempt $attempt/$max_attempts: CIDR $cidr is in state: $state.
 Waiting..."
        sleep 10
        ((attempt++))
    done

    echo "Timed out waiting for CIDR $cidr to be provisioned"
    return 1
}

# Step 1: Create an IPAM
echo "Creating IPAM..."
IPAM_RESULT=$(aws ec2 create-ipam \
    --description "My IPAM" \
    --operating-regions RegionName=us-east-1 RegionName=us-west-2)

if [ $? -ne 0 ]; then
    handle_error "Failed to create IPAM"
fi

IPAM_ID=$(echo "$IPAM_RESULT" | grep -o '"IpamId": "[^"]*' | cut -d'"' -f4)
echo "IPAM created with ID: $IPAM_ID"
```

```
# Wait for IPAM to be created and available
echo "Waiting for IPAM to be available..."
sleep 20

# Step 2: Get the IPAM Scope ID - FIXED to correctly identify the private scope
echo "Getting IPAM Scope ID..."
SCOPE_RESULT=$(aws ec2 describe-ipams --ipam-id "$IPAM_ID")

if [ $? -ne 0 ]; then
    handle_error "Failed to get IPAM details"
fi

# Extract the private scope ID directly from the IPAM details
PRIVATE_SCOPE_ID=$(echo "$SCOPE_RESULT" | grep -o '"PrivateDefaultScopeId": "[^"]*'
 | cut -d'"' -f4)
echo "Private Scope ID: $PRIVATE_SCOPE_ID"

if [ -z "$PRIVATE_SCOPE_ID" ]; then
    handle_error "Failed to get Private Scope ID"
fi

# Step 3: Create a Top-Level IPv4 Pool
echo "Creating Top-level IPv4 Pool..."
TOP_POOL_RESULT=$(aws ec2 create-ipam-pool \
    --ipam-scope-id "$PRIVATE_SCOPE_ID" \
    --address-family ipv4 \
    --description "Top-level pool")

if [ $? -ne 0 ]; then
    handle_error "Failed to create Top-level Pool"
fi

TOP_POOL_ID=$(echo "$TOP_POOL_RESULT" | grep -o '"IpamPoolId": "[^"]*' | cut -d'"' -
f4)
echo "Top-level Pool created with ID: $TOP_POOL_ID"

# Wait for the top-level pool to be available
if ! wait_for_pool "$TOP_POOL_ID"; then
    handle_error "Top-level Pool did not become available in time"
fi

# Provision CIDR to the top-level pool
echo "Provisioning CIDR to Top-level Pool..."
TOP_POOL_CIDR="10.0.0.0/8"
```

```
PROVISION_RESULT=$(aws ec2 provision-ipam-pool-cidr \
    --ipam-pool-id "$TOP_POOL_ID" \
    --cidr "$TOP_POOL_CIDR")

if [ $? -ne 0 ]; then
    handle_error "Failed to provision CIDR to Top-level Pool"
fi

echo "$PROVISION_RESULT"

# Wait for the CIDR to be fully provisioned
if ! wait_for_cidr_provisioning "$TOP_POOL_ID" "$TOP_POOL_CIDR"; then
    handle_error "CIDR provisioning to Top-level Pool did not complete in time"
fi

# Step 4: Create a Regional IPv4 Pool
echo "Creating Regional IPv4 Pool..."
REGIONAL_POOL_RESULT=$(aws ec2 create-ipam-pool \
    --ipam-scope-id "$PRIVATE_SCOPE_ID" \
    --source-ipam-pool-id "$TOP_POOL_ID" \
    --locale us-east-1 \
    --address-family ipv4 \
    --description "Regional pool in us-east-1")

if [ $? -ne 0 ]; then
    handle_error "Failed to create Regional Pool"
fi

REGIONAL_POOL_ID=$(echo "$REGIONAL_POOL_RESULT" | grep -o '"IpamPoolId": "[^"]*' |
 cut -d'"' -f4)
echo "Regional Pool created with ID: $REGIONAL_POOL_ID"

# Wait for the regional pool to be available
if ! wait_for_pool "$REGIONAL_POOL_ID"; then
    handle_error "Regional Pool did not become available in time"
fi

# Provision CIDR to the regional pool
echo "Provisioning CIDR to Regional Pool..."
REGIONAL_POOL_CIDR="10.0.0.0/16"
PROVISION_RESULT=$(aws ec2 provision-ipam-pool-cidr \
    --ipam-pool-id "$REGIONAL_POOL_ID" \
    --cidr "$REGIONAL_POOL_CIDR")
```

```
if [ $? -ne 0 ]; then
    handle_error "Failed to provision CIDR to Regional Pool"
fi

echo "$PROVISION_RESULT"

# Wait for the CIDR to be fully provisioned
if ! wait_for_cidr_provisioning "$REGIONAL_POOL_ID" "$REGIONAL_POOL_CIDR"; then
    handle_error "CIDR provisioning to Regional Pool did not complete in time"
fi

# Step 5: Create a Development IPv4 Pool - FIXED to include locale
echo "Creating Development IPv4 Pool..."
DEV_POOL_RESULT=$(aws ec2 create-ipam-pool \
    --ipam-scope-id "$PRIVATE_SCOPE_ID" \
    --source-ipam-pool-id "$REGIONAL_POOL_ID" \
    --locale us-east-1 \
    --address-family ipv4 \
    --description "Development pool")

if [ $? -ne 0 ]; then
    handle_error "Failed to create Development Pool"
fi

DEV_POOL_ID=$(echo "$DEV_POOL_RESULT" | grep -o '"IpamPoolId": "[^"]*' | cut -d'"' -
f4)
echo "Development Pool created with ID: $DEV_POOL_ID"

# Wait for the development pool to be available
if ! wait_for_pool "$DEV_POOL_ID"; then
    handle_error "Development Pool did not become available in time"
fi

# Provision CIDR to the development pool
echo "Provisioning CIDR to Development Pool..."
DEV_POOL_CIDR="10.0.0.0/24"
PROVISION_RESULT=$(aws ec2 provision-ipam-pool-cidr \
    --ipam-pool-id "$DEV_POOL_ID" \
    --cidr "$DEV_POOL_CIDR")

if [ $? -ne 0 ]; then
    handle_error "Failed to provision CIDR to Development Pool"
fi
```

```
echo "$PROVISION_RESULT"

# Wait for the CIDR to be fully provisioned
if ! wait_for_cidr_provisioning "$DEV_POOL_ID" "$DEV_POOL_CIDR"; then
    handle_error "CIDR provisioning to Development Pool did not complete in time"
fi

# Step 6: Create a VPC Using an IPAM Pool CIDR - FIXED to use the correct parameter
 names and a smaller netmask length
echo "Creating VPC using IPAM Pool CIDR..."
VPC_RESULT=$(aws ec2 create-vpc \
    --ipv4-ipam-pool-id "$DEV_POOL_ID" \
    --ipv4-netmask-length 26 \
    --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=IPAM-VPC}]')

if [ $? -ne 0 ]; then
    handle_error "Failed to create VPC"
fi

VPC_ID=$(echo "$VPC_RESULT" | grep -o '"VpcId": "[^"]*' | cut -d'"' -f4)
echo "VPC created with ID: $VPC_ID"

# Step 7: Verify the IPAM Pool Allocation
echo "Verifying IPAM Pool Allocation..."
ALLOCATION_RESULT=$(aws ec2 get-ipam-pool-allocations \
    --ipam-pool-id "$DEV_POOL_ID")

if [ $? -ne 0 ]; then
    handle_error "Failed to verify IPAM Pool Allocation"
fi

echo "IPAM Pool Allocation verified:"
echo "$ALLOCATION_RESULT" | grep -A 5 "Allocations"

echo ""
echo "IPAM setup completed successfully!"
echo ""

# Prompt for cleanup
cleanup_resources

echo "Script completed at $(date)"
exit 0
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateIpam
  - CreateIpamPool
  - CreateVpc
  - DeleteIpam
  - DeleteIpamPool
  - DeleteVpc
  - DeprovisionIpamPoolCidr
  - DescribeIpamPools
  - DescribeIpams
  - DescribeVpcs
  - GetIpamPoolAllocations
  - GetIpamPoolCidrs
  - ProvisionIpamPoolCidr

HealthImaging exemples d'utilisation AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with HealthImaging.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`CreateDatastore`

L'exemple de code suivant montre comment utiliser`CreateDatastore`.

AWS CLI avec le script Bash

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function imaging_create_datastore
#
# This function creates an AWS HealthImaging data store for importing DICOM P10
 files.
#
# Parameters:
#       -n data_store_name - The name of the data store.
#
# Returns:
#       The datastore ID.
#    And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function imaging_create_datastore() {
  local datastore_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function imaging_create_datastore"
    echo "Creates an AWS HealthImaging data store for importing DICOM P10 files."
    echo "  -n data_store_name - The name of the data store."
    echo ""
  }
```

```
  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) datastore_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$datastore_name" ]]; then
    errecho "ERROR: You must provide a data store name with the -n parameter."
    usage
    return 1
  fi

  response=$(aws medical-imaging create-datastore \
    --datastore-name "$datastore_name" \
    --output text \
    --query 'datastoreId')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports medical-imaging create-datastore operation failed.
$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDatastoreà la section Référence des AWS CLI commandes.

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

`DeleteDatastore`

L'exemple de code suivant montre comment utiliser`DeleteDatastore`.

AWS CLI avec le script Bash

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function imaging_delete_datastore
#
# This function deletes an AWS HealthImaging data store.
#
# Parameters:
#       -i datastore_id - The ID of the data store.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function imaging_delete_datastore() {
  local datastore_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
```

```
    echo "function imaging_delete_datastore"
    echo "Deletes an AWS HealthImaging data store."
    echo "  -i datastore_id - The ID of the data store."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) datastore_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$datastore_id" ]]; then
    errecho "ERROR: You must provide a data store ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws medical-imaging delete-datastore \
    --datastore-id "$datastore_id")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports medical-imaging delete-datastore operation failed.
$response"
    return 1
  fi

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDatastoreà la section Référence des AWS CLI commandes.

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

`GetDatastore`

L'exemple de code suivant montre comment utiliser`GetDatastore`.

AWS CLI avec le script Bash

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function imaging_get_datastore
#
# Get a data store's properties.
#
# Parameters:
#       -i data_store_id - The ID of the data store.
#
# Returns:
#       [datastore_name, datastore_id, datastore_status, datastore_arn,  created_at,
 updated_at]
#    And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function imaging_get_datastore() {
```

```
 local datastore_id option OPTARG # Required to use getopts command in a function.
 local error_code
 # bashsupport disable=BP5008
 function usage() {
   echo "function imaging_get_datastore"
   echo "Gets a data store's properties."
   echo "  -i datastore_id - The ID of the data store."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "i:h" option; do
   case "${option}" in
     i) datastore_id="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$datastore_id" ]]; then
   errecho "ERROR: You must provide a data store ID with the -i parameter."
   usage
   return 1
 fi

 local response

 response=$(
   aws medical-imaging get-datastore \
     --datastore-id "$datastore_id" \
     --output text \
     --query "[ datastoreProperties.datastoreName,
datastoreProperties.datastoreId, datastoreProperties.datastoreStatus,
datastoreProperties.datastoreArn,  datastoreProperties.createdAt,
datastoreProperties.updatedAt]"
 )
```

```
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-datastores operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous GetDatastoreà la section Référence des AWS CLI commandes.

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

`ListDatastores`

L'exemple de code suivant montre comment utiliser`ListDatastores`.

AWS CLI avec le script Bash

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function imaging_list_datastores
#
# List the HealthImaging data stores in the account.
```

```
#
# Returns:
#       [[datastore_name, datastore_id, datastore_status]]
#    And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function imaging_list_datastores() {
  local option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function imaging_list_datastores"
    echo "Lists the AWS HealthImaging data stores in the account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response
  response=$(aws medical-imaging list-datastores \
    --output text \
    --query "datastoreSummaries[*][datastoreName, datastoreId, datastoreStatus]")
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-datastores operation failed.$response"
    return 1
  fi
```

```
  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous ListDatastoresà la section Référence des AWS CLI commandes.

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exemples d'utilisation d'IAM AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash avec IAM.

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Principes de base
- Actions
- Scénarios

Principes de base

Principes de base

L'exemple de code suivant montre comment créer un utilisateur et assumer un rôle.

> **Warning**
>
> Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que AWS IAM Identity Center.

- Créer un utilisateur sans autorisation.
- Créer un rôle qui accorde l'autorisation de répertorier les compartiments Amazon S3 pour le compte.
- Ajouter une politique pour permettre à l'utilisateur d'assumer le rôle.
- Assumez le rôle et répertorier les compartiments S3 à l'aide d'informations d'identification temporaires, puis nettoyez les ressources.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function iam_create_user_assume_role
#
# Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
#
#     "IAM access" permissions are needed to run this code.
#     "STS assume role" permissions are needed to run this code. (Note: It might be
 necessary to
#           create a custom policy).
```

```
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function iam_create_user_assume_role() {
  {
    if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

      source ./iam_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the IAM create user and assume role demo."
  echo
  echo "This demo will create an IAM user, create an IAM role, and apply the role to
 the user."
  echo_repeat "*" 88
  echo

  echo -n "Enter a name for a new IAM user: "
  get_input
  user_name=$get_input_result

  local user_arn
  user_arn=$(iam_create_user -u "$user_name")

  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created demo IAM user named $user_name"
  else
    errecho "$user_arn"
    errecho "The user failed to create. This demo will exit."
    return 1
  fi

  local access_key_response
  access_key_response=$(iam_create_user_access_key -u "$user_name")
  # shellcheck disable=SC2181
  if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
    return 1
```

```
  fi

  IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
  local key_name=${access_key_values[0]}
  local key_secret=${access_key_values[1]}

  echo "Created access key named $key_name"

  echo "Wait 10 seconds for the user to be ready."
  sleep 10
  echo_repeat "*" 88
  echo

  local iam_role_name
  iam_role_name=$(generate_random_name "test-role")
  echo "Creating a role named $iam_role_name with user $user_name as the principal."

  local assume_role_policy_document="{
    \"Version\": \"2012-10-17\",
    \"Statement\": [{
        \"Effect\": \"Allow\",
        \"Principal\": {\"AWS\": \"$user_arn\"},
        \"Action\": \"sts:AssumeRole\"
        }]
    }"

  local role_arn
  role_arn=$(iam_create_role -n "$iam_role_name" -p "$assume_role_policy_document")

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created IAM role named $iam_role_name"
  else
    errecho "The role failed to create. This demo will exit."
    clean_up "$user_name" "$key_name"
    return 1
  fi

  local policy_name
  policy_name=$(generate_random_name "test-policy")
  local policy_document="{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
```

```
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]}"

  local policy_arn
  policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created  IAM policy named $policy_name"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name"
    return 1
  fi

  if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
    echo "Attached policy $policy_arn to role $iam_role_name"
  else
    errecho "The policy failed to attach."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
    return 1
  fi

  local assume_role_policy_document="{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"$role_arn\"}]}"

  local assume_role_policy_name
  assume_role_policy_name=$(generate_random_name "test-assume-role-")

  # shellcheck disable=SC2181
  local assume_role_policy_arn
  assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created  IAM policy named $assume_role_policy_name for sts assume role"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
    return 1
  fi
```

```
 echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
 sleep 10
 echo_repeat "*" 88
 echo

 echo "Try to list buckets without the new user assuming the role."
 echo_repeat "*" 88
 echo

 # Set the environment variables for the created user.
 # bashsupport disable=BP2001
 export AWS_ACCESS_KEY_ID=$key_name
 # bashsupport disable=BP2001
 export AWS_SECRET_ACCESS_KEY=$key_secret

 local buckets
 buckets=$(s3_list_buckets)

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   local bucket_count
   bucket_count=$(echo "$buckets" | wc -w | xargs)
   echo "There are $bucket_count buckets in the account. This should not have
happened."
 else
   errecho "Because the role with permissions has not been assumed, listing buckets
failed."
 fi

 echo
 echo_repeat "*" 88
 echo "Now assume the role $iam_role_name and list the buckets."
 echo_repeat "*" 88
 echo

 local credentials

 credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Assumed role $iam_role_name"
 else
```

```
    errecho "Failed to assume role."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
  fi

  IFS=$'\t ' read -r -a credentials <<<"$credentials"

  export AWS_ACCESS_KEY_ID=${credentials[0]}
  export AWS_SECRET_ACCESS_KEY=${credentials[1]}
  # bashsupport disable=BP2001
  export AWS_SESSION_TOKEN=${credentials[2]}

  buckets=$(s3_list_buckets)

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets succeeded
because of "
    echo "the assumed role."
  else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
  fi

  local result=0
  export AWS_ACCESS_KEY_ID=""
  export AWS_SECRET_ACCESS_KEY=""

  echo
  echo_repeat "*" 88
  echo "The created resources will now be deleted."
  echo_repeat "*" 88
  echo
```

```
  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    result=1
  fi

  return $result
}
```

Fonctions IAM utilisées dans ce scénario.

```
##############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
##############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
```

```
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}

###############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
```

```
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

```
###############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
```

```
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
```

```
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
```

```
  echo "Creates an AWS Identity and Access Management (IAM) policy."
  echo "  -n policy_name   The name of the IAM policy."
  echo "  -p policy_json -- The policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) policy_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
  errecho "ERROR: You must provide a policy name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-policy \
  --policy-name "$policy_name" \
  --policy-document "$policy_document" \
  --output text \
  --query Policy.Arn)

local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

########################################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
```

```
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
```

```
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
```

```
  echo "Deletes an AWS Identity and Access Management (IAM) policy"
  echo "  -n policy_arn -- The name of the IAM policy arn."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy arn with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    Policy arn:  $policy_arn"
iecho ""

response=$(aws iam delete-policy \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
  return 1
fi

iecho "delete-policy response:$response"
```

```
  iecho

  return 0
}

################################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
```

```
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an AWS Identity and Access Management (IAM) access key for the
specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
```

```
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user."
```

```
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    User name:   $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
  errecho "ERROR: A user with that name does not exist in the account."
  return 1
fi

response=$(aws iam delete-user \
  --user-name "$user_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-user operation failed.$response"
```

```
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Actions

`AttachRolePolicy`

L'exemple de code suivant montre comment utiliser`AttachRolePolicy`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
```

```
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam attach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
  return 1
fi
```

```
  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous AttachRolePolicyà la section Référence des AWS CLI commandes.

`CreateAccessKey`

L'exemple de code suivant montre comment utiliser`CreateAccessKey`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
```

```
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)
```

```
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAccessKeyà la section Référence des AWS CLI commandes.

`CreatePolicy`

L'exemple de code suivant montre comment utiliser`CreatePolicy`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
```

```
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}
```

- Pour plus de détails sur l'API, reportez-vous CreatePolicyà la section Référence des AWS CLI commandes.

`CreateRole`

L'exemple de code suivant montre comment utiliser`CreateRole`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
```

```
  echo "Creates an AWS Identity and Access Management (IAM) role."
  echo "  -n role_name   The name of the IAM role."
  echo "  -p policy_json -- The assume role policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-role \
  --role-name "$role_name" \
  --assume-role-policy-document "$policy_document" \
  --output text \
  --query Role.Arn)

local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateRoleà la section Référence des AWS CLI commandes.

`CreateUser`

L'exemple de code suivant montre comment utiliser`CreateUser`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
```

```
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous CreateUserà la section Référence des AWS CLI commandes.

`DeleteAccessKey`

L'exemple de code suivant montre comment utiliser`DeleteAccessKey`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
```

```
 function usage() {
   echo "function iam_delete_access_key"
   echo "Deletes an AWS Identity and Access Management (IAM) access key for the
specified IAM user"
   echo "  -u user_name    The name of the user."
   echo "  -k access_key   The access key to delete."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "u:k:h" option; do
   case "${option}" in
     u) user_name="${OPTARG}" ;;
     k) access_key="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$user_name" ]]; then
   errecho "ERROR: You must provide a username with the -u parameter."
   usage
   return 1
 fi

 if [[ -z "$access_key" ]]; then
   errecho "ERROR: You must provide an access key with the -k parameter."
   usage
   return 1
 fi

 iecho "Parameters:\n"
 iecho "    Username:   $user_name"
 iecho "    Access key:   $access_key"
 iecho ""
```

```
  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAccessKeyà la section Référence des AWS CLI commandes.

`DeletePolicy`

L'exemple de code suivant montre comment utiliser`DeletePolicy`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
```

```
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an AWS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
```

```
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeletePolicyà la section Référence des AWS CLI commandes.

`DeleteRole`

L'exemple de code suivant montre comment utiliser`DeleteRole`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
```

```
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")
```

```
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteRoleà la section Référence des AWS CLI commandes.

`DeleteUser`

L'exemple de code suivant montre comment utiliser`DeleteUser`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
```

```
}

########################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
########################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

########################################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
########################################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteUserà la section Référence des AWS CLI commandes.

`DetachRolePolicy`

L'exemple de code suivant montre comment utiliser`DetachRolePolicy`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
```

```
 function usage() {
   echo "function iam_detach_role_policy"
   echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
   echo "  -n role_name   The name of the IAM role."
   echo "  -p policy_ARN -- The IAM policy document ARN."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "n:p:h" option; do
   case "${option}" in
     n) role_name="${OPTARG}" ;;
     p) policy_arn="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$role_name" ]]; then
   errecho "ERROR: You must provide a role name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$policy_arn" ]]; then
   errecho "ERROR: You must provide a policy ARN with the -p parameter."
   usage
   return 1
 fi

 response=$(aws iam detach-role-policy \
   --role-name "$role_name" \
   --policy-arn "$policy_arn")

 local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous DetachRolePolicyà la section Référence des AWS CLI commandes.

`GetUser`

L'exemple de code suivant montre comment utiliser`GetUser`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_user_exists
#
```

```
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous GetUserà la section Référence des AWS CLI commandes.

`ListAccessKeys`

L'exemple de code suivant montre comment utiliser`ListAccessKeys`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_list_access_keys
#
# This function lists the access keys for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#
# Returns:
#       access_key_ids
#     And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function iam_list_access_keys() {

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_access_keys"
    echo "Lists the AWS Identity and Access Management (IAM) access key IDs for the
 specified user."
    echo "  -u user_name   The name of the IAM user."
    echo ""
  }
```

```
  local user_name response
  local option OPTARG # Required to use getopts command in a function.
  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam list-access-keys \
    --user-name "$user_name" \
    --output text \
    --query 'AccessKeyMetadata[].AccessKeyId')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-access-keys operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous ListAccessKeysà la section Référence des AWS CLI commandes.

`ListUsers`

L'exemple de code suivant montre comment utiliser`ListUsers`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_list_users
#
# List the IAM users in the account.
#
# Returns:
#       The list of users names
#    And:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
##############################################################################
function iam_list_users() {
  local option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_users"
    echo "Lists the AWS Identity and Access Management (IAM) user in the account."
```

```
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws iam list-users \
    --output text \
    --query "Users[].UserName")
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-users operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous ListUsersà la section Référence des AWS CLI commandes.

`UpdateAccessKey`

L'exemple de code suivant montre comment utiliser`UpdateAccessKey`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function iam_update_access_key
#
# This function can activate or deactivate an IAM access key for the specified IAM
 user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to update.
#       -a            -- Activate the selected access key.
#       -d            -- Deactivate the selected access key.
#
# Example:
#       # To deactivate the selected access key for IAM user Bob
#       iam_update_access_key -u Bob -k AKIAIOSFODNN7EXAMPLE -d
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_update_access_key() {
  local user_name access_key status response
  local option OPTARG # Required to use getopts command in a function.
  local activate_flag=false deactivate_flag=false

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_update_access_key"
    echo "Updates the status of an AWS Identity and Access Management (IAM) access
 key for the specified IAM user"
    echo "  -u user_name    The name of the user."
```

```
    echo "  -k access_key   The access key to update."
    echo "  -a              Activate the access key."
    echo "  -d              Deactivate the access key."
    echo ""
  }

  # Retrieve the calling parameters.
    while getopts "u:k:adh" option; do
      case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        a) activate_flag=true ;;
        d) deactivate_flag=true ;;
        h)
          usage
          return 0
          ;;
        \?)
          echo "Invalid parameter"
          usage
          return 1
          ;;
      esac
    done
    export OPTIND=1

   # Validate input parameters
    if [[ -z "$user_name" ]]; then
      errecho "ERROR: You must provide a username with the -u parameter."
      usage
      return 1
    fi

    if [[ -z "$access_key" ]]; then
      errecho "ERROR: You must provide an access key with the -k parameter."
      usage
      return 1
    fi

    # Ensure that only -a or -d is specified
    if [[ "$activate_flag" == true && "$deactivate_flag" == true ]]; then
      errecho "ERROR: You cannot specify both -a (activate) and -d (deactivate) at
the same time."
      usage
```

```
      return 1
    fi

    # If neither -a nor -d is provided, return an error
    if [[ "$activate_flag" == false && "$deactivate_flag" == false ]]; then
      errecho "ERROR: You must specify either -a (activate) or -d (deactivate)."
      usage
      return 1
    fi

    # Determine the status based on the flag
    if [[ "$activate_flag" == true ]]; then
      status="Active"
    elif [[ "$deactivate_flag" == true ]]; then
      status="Inactive"
    fi

    iecho "Parameters:\n"
    iecho "    Username:   $user_name"
    iecho "    Access key: $access_key"
    iecho "    New status: $status"
    iecho ""

    # Update the access key status
    response=$(aws iam update-access-key \
      --user-name "$user_name" \
      --access-key-id "$access_key" \
      --status "$status" 2>&1)

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log $error_code
      errecho "ERROR: AWS reports update-access-key operation failed.\n$response"
      return 1
    fi

    iecho "update-access-key response: $response"
    iecho

    return 0
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateAccessKeyà la section Référence des AWS CLI commandes.

Scénarios

Configurer le contrôle d'accès basé sur les attributs

L'exemple de code suivant montre comment implémenter le contrôle d'accès basé sur les attributs (ABAC) pour DynamoDB.

- Créez une politique IAM pour ABAC.
- Créez des tableaux avec des balises pour différents départements.
- Répertoriez et filtrez les tableaux en fonction des balises.

AWS CLI avec le script Bash

Créez une politique IAM pour ABAC.

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

Créez des tableaux avec des balises pour différents départements.

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
```

```
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
        Key=Environment,Value=Production
```

Répertoriez et filtrez les tableaux en fonction des balises.

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreatePolicy

- CreateTable
- ListTables

Travaillez avec Streams et Time-to-Live

L'exemple de code suivant montre comment gérer les flux et les fonctionnalités de DynamoDB. Time-to-Live

- Créez une table avec Streams activé.
- Décrivez les flux.
- Créez une fonction Lambda pour le traitement des flux.
- Activez le TTL sur une table.
- Ajoutez des éléments dotés d'attributs TTL.
- Décrivez les paramètres TTL.

AWS CLI avec le script Bash

Créez une table avec Streams activé.

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

Décrivez les flux.

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
```

```
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"

# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

Créez une fonction Lambda pour Streams.

```
# Step 1: Create an IAM role for the Lambda function
cat > trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
```

```
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

Activez le TTL sur une table.

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST

# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo

# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

Ajoutez des éléments dotés d'attributs TTL.

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
```

```
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

Décrivez les paramètres TTL.

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AttachRolePolicy
  - CreateRole
  - CreateTable
  - DescribeTable
  - DescribeTimeToLive
  - PutItem
  - UpdateTimeToLive

AWS KMS exemples d'utilisation AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with AWS KMS.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Scénarios

Scénarios

Travaillez avec le chiffrement des tables

L'exemple de code suivant montre comment gérer les options de chiffrement pour les tables DynamoDB.

- Créez une table avec le chiffrement par défaut.
- Créez une table avec une clé CMK gérée par le client.
- Mettez à jour les paramètres de chiffrement des tables.
- Décrivez le chiffrement des tables.

AWS CLI avec le script Bash

Créez une table avec le chiffrement par défaut.

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

Créez une table avec une clé CMK gérée par le client.

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
```

```
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)

# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Mettez à jour le chiffrement des tables.

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Décrivez le chiffrement des tables.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

Exemples AWS CLI d'utilisation de Lightsail avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with Lightsail.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Scénarios

Scénarios

Commencez avec Lightsail

L'exemple de code suivant illustre comment :

- Explorez les plans et les offres groupées disponibles pour la création d'instances
- Création d'une instance Lightsail avec Amazon Linux 2023
- Surveiller l'état de l'instance jusqu'à son exécution
- Télécharger la paire de clés par défaut pour l'accès SSH
- Création et connexion d'un disque de stockage par blocs supplémentaire
- Création d'un instantané d'instance à des fins de sauvegarde
- Nettoyez les ressources pour éviter les frais récurrents

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash

# Amazon Lightsail Getting Started CLI Script
# This script demonstrates how to create and manage Lightsail resources using the
 AWS CLI

# FIXES APPLIED:
# 1. Added polling mechanism to check disk state before attaching
# 2. Added polling mechanism to check snapshot state before proceeding with cleanup
# 3. Set AWS_REGION variable to us-west-2 for consistent region usage

# Set AWS region
export AWS_REGION="us-west-2"
echo "Using AWS region: $AWS_REGION"

# Set up logging
LOG_FILE="lightsail-script.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Lightsail Getting Started script at $(date)"

# Error handling function
handle_error() {
  echo "ERROR: $1"
  echo "Attempting to clean up resources..."
  cleanup_resources
  exit 1
}

# Function to check if a command succeeded
check_status() {
  if [ $? -ne 0 ]; then
    handle_error "$1"
  fi
}

# Generate a random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
INSTANCE_NAME="LightsailInstance-${RANDOM_ID}"
DISK_NAME="LightsailDisk-${RANDOM_ID}"
SNAPSHOT_NAME="LightsailSnapshot-${RANDOM_ID}"

# Array to track created resources
```

```
declare -a CREATED_RESOURCES

# Function to add a resource to the tracking array
track_resource() {
  CREATED_RESOURCES+=("$1:$2")
  echo "Created $1: $2"
}

# Function to clean up resources
cleanup_resources() {
  echo "Resources created by this script:"
  for resource in "${CREATED_RESOURCES[@]}"; do
    echo "  $resource"
  done

  # Reverse the array to delete resources in reverse order
  for (( idx=${#CREATED_RESOURCES[@]}-1 ; idx>=0 ; idx-- )); do
    IFS=':' read -r type name <<< "${CREATED_RESOURCES[idx]}"

    case "$type" in
      "instance_snapshot")
        echo "Deleting instance snapshot: $name"
        aws lightsail delete-instance-snapshot --instance-snapshot-name "$name" --
region $AWS_REGION
        ;;
      "disk_snapshot")
        echo "Deleting disk snapshot: $name"
        aws lightsail delete-disk-snapshot --disk-snapshot-name "$name" --region
 $AWS_REGION
        ;;
      "disk")
        echo "Detaching disk: $name"
        aws lightsail detach-disk --disk-name "$name" --region $AWS_REGION
        sleep 10 # Wait for detach to complete
        echo "Deleting disk: $name"
        aws lightsail delete-disk --disk-name "$name" --region $AWS_REGION
        ;;
      "instance")
        echo "Deleting instance: $name"
        # Check instance state before attempting to delete
        INSTANCE_STATE=$(aws lightsail get-instance-state --instance-name "$name" --
region $AWS_REGION --query 'state.name' --output text 2>/dev/null)
        if [ "$INSTANCE_STATE" == "pending" ]; then
```

```
          echo "Instance is in pending state. Waiting for it to be ready before
 deleting..."
          MAX_WAIT=30
          WAITED=0
          while [ "$INSTANCE_STATE" == "pending" ] && [ $WAITED -lt $MAX_WAIT ]; do
            sleep 10
            WAITED=$((WAITED+1))
            INSTANCE_STATE=$(aws lightsail get-instance-state --instance-name
 "$name" --region $AWS_REGION --query 'state.name' --output text 2>/dev/null)
            echo "Instance state: $INSTANCE_STATE"
          done
        fi
        aws lightsail delete-instance --instance-name "$name" --region $AWS_REGION
        ;;
    esac
  done

  echo "Cleanup completed"
}

# Step 1: Verify AWS CLI configuration
echo "Step 1: Verifying AWS CLI configuration"
aws configure list
check_status "Failed to verify AWS CLI configuration"

# Step 2: Get available blueprints and bundles
echo "Step 2: Getting available blueprints and bundles"
echo "Available blueprints (showing first 5):"
aws lightsail get-blueprints --region $AWS_REGION --query 'blueprints[0:5].
[blueprintId,name]' --output table
check_status "Failed to get blueprints"

echo "Available bundles (showing first 5):"
aws lightsail get-bundles --region $AWS_REGION --query 'bundles[0:5].
[bundleId,name,price]' --output table
check_status "Failed to get bundles"

# Get available regions and availability zones
echo "Getting available regions and availability zones"
# Use a specific availability zone in us-west-2 region
AVAILABILITY_ZONE="us-west-2a"
echo "Using availability zone: $AVAILABILITY_ZONE"

# Step 3: Create a Lightsail instance
```

```
echo "Step 3: Creating Lightsail instance: $INSTANCE_NAME"
aws lightsail create-instances \
  --instance-names "$INSTANCE_NAME" \
  --availability-zone "$AVAILABILITY_ZONE" \
  --blueprint-id amazon_linux_2023 \
  --bundle-id nano_3_0 \
  --region $AWS_REGION
check_status "Failed to create Lightsail instance"
track_resource "instance" "$INSTANCE_NAME"

# Wait for the instance to be in a running state
echo "Waiting for instance to be in running state..."
# Wait for the instance to be ready (polling approach)
MAX_ATTEMPTS=30
ATTEMPTS=0
while [ $ATTEMPTS -lt $MAX_ATTEMPTS ]; do
  STATUS=$(aws lightsail get-instance-state --instance-name "$INSTANCE_NAME" --
region $AWS_REGION --query 'state.name' --output text)
  if [ "$STATUS" == "running" ]; then
    echo "Instance is now running"
    break
  fi
  echo "Instance status: $STATUS. Waiting..."
  ATTEMPTS=$((ATTEMPTS+1))
  sleep 10
done

if [ $ATTEMPTS -eq $MAX_ATTEMPTS ]; then
  handle_error "Instance failed to reach running state after 5 minutes"
fi

# Get instance details
echo "Getting instance details"
INSTANCE_IP=$(aws lightsail get-instance --instance-name "$INSTANCE_NAME" --region
 $AWS_REGION --query 'instance.publicIpAddress' --output text)
check_status "Failed to get instance IP address"
echo "Instance IP address: $INSTANCE_IP"

# Step 4: Download the default key pair
echo "Step 4: Downloading default key pair"
KEY_FILE="lightsail_key_${RANDOM_ID}.pem"
aws lightsail download-default-key-pair --region $AWS_REGION --output text >
 "$KEY_FILE"
check_status "Failed to download key pair"
```

```
chmod 400 "$KEY_FILE"
check_status "Failed to set permissions on key pair"
echo "Key pair downloaded to $KEY_FILE"

echo "To connect to your instance, use:"
echo "ssh -i $KEY_FILE ec2-user@$INSTANCE_IP"

# Step 5: Create a block storage disk
echo "Step 5: Creating block storage disk: $DISK_NAME"
aws lightsail create-disk \
  --disk-name "$DISK_NAME" \
  --availability-zone "$AVAILABILITY_ZONE" \
  --size-in-gb 8 \
  --region $AWS_REGION
check_status "Failed to create disk"
track_resource "disk" "$DISK_NAME"

# FIX: Wait for the disk to be available using polling instead of fixed sleep
echo "Waiting for disk to be available..."
MAX_ATTEMPTS=30
ATTEMPTS=0
while [ $ATTEMPTS -lt $MAX_ATTEMPTS ]; do
  DISK_STATE=$(aws lightsail get-disk --disk-name "$DISK_NAME" --region $AWS_REGION
 --query 'disk.state' --output text 2>/dev/null)
  if [ "$DISK_STATE" == "available" ]; then
    echo "Disk is now available"
    break
  fi
  echo "Disk status: $DISK_STATE. Waiting..."
  ATTEMPTS=$((ATTEMPTS+1))
  sleep 10
done

if [ $ATTEMPTS -eq $MAX_ATTEMPTS ]; then
  handle_error "Disk failed to become available after 5 minutes"
fi

# Attach the disk to the instance
echo "Attaching disk to instance"
aws lightsail attach-disk \
  --disk-name "$DISK_NAME" \
  --instance-name "$INSTANCE_NAME" \
  --disk-path /dev/xvdf \
  --region $AWS_REGION
```

```
check_status "Failed to attach disk to instance"

echo "Disk attached. To format and mount the disk, connect to your instance and
 run:"
echo "sudo mkfs -t ext4 /dev/xvdf"
echo "sudo mkdir -p /mnt/my-data"
echo "sudo mount /dev/xvdf /mnt/my-data"
echo "sudo chown ec2-user:ec2-user /mnt/my-data"

# Step 6: Create a snapshot of the instance
echo "Step 6: Creating snapshot of the instance: $SNAPSHOT_NAME"
aws lightsail create-instance-snapshot \
  --instance-name "$INSTANCE_NAME" \
  --instance-snapshot-name "$SNAPSHOT_NAME" \
  --region $AWS_REGION
check_status "Failed to create instance snapshot"
track_resource "instance_snapshot" "$SNAPSHOT_NAME"

# FIX: Wait for the snapshot to complete using polling instead of fixed sleep
echo "Waiting for snapshot to complete... (this may take several minutes)"
MAX_ATTEMPTS=60  # Increased timeout for snapshot creation
ATTEMPTS=0
while [ $ATTEMPTS -lt $MAX_ATTEMPTS ]; do
  SNAPSHOT_STATE=$(aws lightsail get-instance-snapshot --instance-snapshot-name
 "$SNAPSHOT_NAME" --region $AWS_REGION --query 'instanceSnapshot.state' --output
 text 2>/dev/null)
  if [ "$SNAPSHOT_STATE" == "completed" ]; then
    echo "Snapshot creation completed"
    break
  fi
  echo "Snapshot status: $SNAPSHOT_STATE. Waiting... ($ATTEMPTS/$MAX_ATTEMPTS)"
  ATTEMPTS=$((ATTEMPTS+1))
  sleep 10
done

if [ $ATTEMPTS -eq $MAX_ATTEMPTS ]; then
  echo "Warning: Snapshot creation is taking longer than expected but will continue
 in the background."
  echo "You can check its status later with: aws lightsail get-instance-snapshot --
instance-snapshot-name $SNAPSHOT_NAME --region $AWS_REGION"
fi

# Step 7: Clean up resources
echo "Step 7: Clean up resources"
```

```
echo "The script has created the following resources:"
for resource in "${CREATED_RESOURCES[@]}"; do
  echo "  $resource"
done

read -p "Do you want to clean up these resources? (y/n): " CLEANUP_CONFIRM
if [[ "$CLEANUP_CONFIRM" == "y" || "$CLEANUP_CONFIRM" == "Y" ]]; then
  cleanup_resources
else
  echo "Resources will not be cleaned up. You can manually delete them later."
  echo "To clean up manually, use the following commands:"
  echo "aws lightsail delete-instance-snapshot --instance-snapshot-name
 $SNAPSHOT_NAME --region $AWS_REGION"
  echo "aws lightsail detach-disk --disk-name $DISK_NAME --region $AWS_REGION"
  echo "aws lightsail delete-disk --disk-name $DISK_NAME --region $AWS_REGION"
  echo "aws lightsail delete-instance --instance-name $INSTANCE_NAME --region
 $AWS_REGION"
fi

echo "Script completed at $(date)"
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - AttachDisk
  - CreateDisk
  - CreateInstanceSnapshot
  - CreateInstances
  - DeleteDisk
  - DeleteInstance
  - DeleteInstanceSnapshot
  - DetachDisk
  - DownloadDefaultKeyPair
  - GetBlueprints
  - GetBundles
  - GetDisk
  - GetInstance

- GetInstanceSnapshot
- GetInstanceState

Exemples d'utilisation d'Amazon S3 AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash avec Amazon S3.

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Principes de base
- Actions
- Scénarios

Principes de base

Principes de base

L'exemple de code suivant illustre comment :

- créer un compartiment et y charger un fichier ;
- télécharger un objet à partir d'un compartiment ;
- copier un objet dans le sous-dossier d'un compartiment ;
- répertorier les objets d'un compartiment ;

- supprimer le compartiment et tous les objets qui y figurent.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function s3_getting_started() {
  {
    if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
      cd bucket-lifecycle-operations || exit

      source ./bucket_operations.sh
      cd ..
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon S3 getting started demo."
  echo_repeat "*" 88
    echo "A unique bucket will be created by appending a Universally Unique
 Identifier to a bucket name prefix."
    echo -n "Enter a prefix for the S3 bucket that will be used in this demo: "
    get_input
    bucket_name_prefix=$get_input_result
  local bucket_name
  bucket_name=$(generate_random_name "$bucket_name_prefix")

  local region_code
  region_code=$(aws configure get region)
```

```
  if create_bucket -b "$bucket_name" -r "$region_code"; then
    echo "Created demo bucket named $bucket_name"
  else
    errecho "The bucket failed to create. This demo will exit."
    return 1
  fi

  local file_name
  while [ -z "$file_name" ]; do
    echo -n "Enter a file you want to upload to your bucket: "
    get_input
    file_name=$get_input_result

    if [ ! -f "$file_name" ]; then
      echo "Could not find file $file_name. Are you sure it exists?"
      file_name=""
    fi
  done

  local key
  key="$(basename "$file_name")"

  local result=0
  if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
    echo "Uploaded file $file_name into bucket $bucket_name with key $key."
  else
    result=1
  fi

  local destination_file
  destination_file="$file_name.download"
  if yes_no_input "Would you like to download $key to the file $destination_file?
 (y/n) "; then
    if download_object_from_bucket "$bucket_name" "$destination_file" "$key"; then
      echo "Downloaded $key in the bucket $bucket_name to the file
 $destination_file."
    else
      result=1
    fi
  fi

  if yes_no_input "Would you like to copy $key a new object key in your bucket? (y/
n) "; then
```

```
    local to_key
    to_key="demo/$key"
    if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
      echo "Copied $key in the bucket $bucket_name to the  $to_key."
    else
      result=1
    fi
  fi

  local bucket_items
  bucket_items=$(list_items_in_bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    result=1
  fi

  echo "Your bucket contains the following items."
  echo -e "Name\t\tSize"
  echo "$bucket_items"

  if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
 (y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
      echo "The following items were deleted from the bucket $bucket_name"
      echo "$bucket_items"
    else
      result=1
    fi

    if delete_bucket "$bucket_name"; then
      echo "Deleted the bucket $bucket_name"
    else
      result=1
    fi
  fi

  return $result
}
```

Les fonctions Amazon S3 utilisées dans ce scénario.

```
###############################################################################
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#       -b bucket_name  -- The name of the bucket to create.
#       -r region_code  -- The code for an AWS Region in which to
#                          create the bucket.
#
# Returns:
#       The URL of the bucket that was created.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function create_bucket() {
  local bucket_name region_code response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi

  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function copy_file_to_bucket
```

```
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

###############################################################################
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
```

```
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response

  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")

  # shellcheck disable=SC2181
```

```
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
```

```
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
}

###############################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
```

```
#       1 - If it fails.
################################################################################
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Actions

`CopyObject`

L'exemple de code suivant montre comment utiliser`CopyObject`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response

  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")
```

```
  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS CLI commandes.

`CreateBucket`

L'exemple de code suivant montre comment utiliser`CreateBucket`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
```

```
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#       -b bucket_name  -- The name of the bucket to create.
#       -r region_code  -- The code for an AWS Region in which to
#                          create the bucket.
#
# Returns:
#       The URL of the bucket that was created.
#     And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function create_bucket() {
  local bucket_name region_code response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi

  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS CLI commandes.

`DeleteBucket`

L'exemple de code suivant montre comment utiliser`DeleteBucket`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")
```

```
  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS CLI commandes.

`DeleteObject`

L'exemple de code suivant montre comment utiliser`DeleteObject`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
########################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
########################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

########################################################################
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - The key (file name) in the bucket to delete.
```

```
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_item_in_bucket() {
  local bucket_name=$1
  local key=$2
  local response

  response=$(aws s3api delete-object \
    --bucket "$bucket_name" \
    --key "$key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS CLI commandes.

`DeleteObjects`

L'exemple de code suivant montre comment utiliser`DeleteObjects`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
```

```
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS CLI commandes.

`GetObject`

L'exemple de code suivant montre comment utiliser`GetObject`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
```

```
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

`HeadBucket`

L'exemple de code suivant montre comment utiliser`HeadBucket`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
################################################################################
# function bucket_exists
#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#       $1 - The name of the bucket to check.
#
# Returns:
#       0 - If the bucket already exists.
#       1 - If the bucket doesn't exist.
```

```
#############################################################################
function bucket_exists() {
  local bucket_name
  bucket_name=$1

  # Check whether the bucket already exists.
  # We suppress all output - we're interested only in the return code.

  if aws s3api head-bucket \
    --bucket "$bucket_name" \
    >/dev/null 2>&1; then
    return 0 # 0 in Bash script means true.
  else
    return 1 # 1 in Bash script means false.
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous HeadBucketà la section Référence des AWS CLI commandes.

`ListObjectsV2`

L'exemple de code suivant montre comment utiliser`ListObjectsV2`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans AWS CLI Command Reference.

`PutObject`

L'exemple de code suivant montre comment utiliser`PutObject`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")
```

```
  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

Scénarios

Commencez avec S3

L'exemple de code suivant illustre comment :

- Création d'un compartiment S3 avec un nom et une configuration régionaux uniques
- Configurer les paramètres de sécurité du bucket, y compris le blocage de l'accès public
- Activez le contrôle des versions et le chiffrement par défaut pour la protection des données
- Importer des objets avec et sans métadonnées personnalisées
- Télécharger des objets depuis le bucket vers le stockage local
- Copiez des objets dans le compartiment pour organiser les données dans des dossiers
- Répertorier le contenu du bucket et les objets avec des préfixes spécifiques
- Ajouter des balises aux compartiments pour la gestion des ressources
- Nettoyez toutes les ressources, y compris les objets versionnés

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et apprenez à le configurer et à l'exécuter dans le référentiel de tutoriels pour développeurs Sample.

```
#!/bin/bash
```

```
# Amazon S3 Getting Started Tutorial Script
# This script demonstrates basic S3 operations including:
# - Creating a bucket
# - Configuring bucket settings
# - Uploading, downloading, and copying objects
# - Deleting objects and buckets

# Latest fixes:
# 1. Fixed folder creation using temporary file
# 2. Corrected versioned object deletion in cleanup
# 3. Improved error handling for cleanup operations

# Set up error handling
set -e
trap 'cleanup_handler $?' EXIT

# Log file setup
LOG_FILE="s3-tutorial-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

# Function to log messages
log() {
    echo "[$(date +"%Y-%m-%d %H:%M:%S")] $1"
}

# Function to handle errors
handle_error() {
    log "ERROR: $1"
    exit 1
}

# Function to check if a bucket exists
bucket_exists() {
    if aws s3api head-bucket --bucket "$1" 2>/dev/null; then
        return 0
    else
        return 1
    fi
}

# Function to delete all versions of objects in a bucket
delete_all_versions() {
    local bucket=$1
    log "Deleting all object versions from bucket $bucket..."
```

```
    # Get and delete all versions
    versions=$(aws s3api list-object-versions --bucket "$bucket" --query
 'Versions[].{Key:Key,VersionId:VersionId}' --output json 2>/dev/null)
    if [ -n "$versions" ] && [ "$versions" != "null" ]; then
        echo "{\"Objects\": $versions}" | aws s3api delete-objects --bucket
 "$bucket" --delete file:///dev/stdin >/dev/null 2>&1 || log "Warning: Some versions
 could not be deleted"
    fi

    # Get and delete all delete markers
    markers=$(aws s3api list-object-versions --bucket "$bucket" --query
 'DeleteMarkers[].{Key:Key,VersionId:VersionId}' --output json 2>/dev/null)
    if [ -n "$markers" ] && [ "$markers" != "null" ]; then
        echo "{\"Objects\": $markers}" | aws s3api delete-objects --bucket "$bucket"
 --delete file:///dev/stdin >/dev/null 2>&1 || log "Warning: Some delete markers
 could not be deleted"
    fi
}

# Function to handle cleanup on exit
cleanup_handler() {
    local exit_code=$1

    # Only run cleanup if it hasn't been run already
    if [ -z "$CLEANUP_DONE" ]; then
        cleanup
    fi

    exit $exit_code
}

# Function to clean up resources
cleanup() {
    log "Starting cleanup process..."
    CLEANUP_DONE=1

    # List all resources created for confirmation
    log "Resources created:"
    if [ -n "$BUCKET_NAME" ]; then
        log "- S3 Bucket: $BUCKET_NAME"

        # Only try to list objects if the bucket exists
        if bucket_exists "$BUCKET_NAME"; then
```

```
            # Check if any objects were created
            OBJECTS=$(aws s3api list-objects-v2 --bucket "$BUCKET_NAME" --query
 'Contents[].Key' --output text 2>/dev/null || echo "")
            if [ -n "$OBJECTS" ]; then
                log "- Objects in bucket:"
                echo "$OBJECTS" | tr '\t' '\n' | while read -r obj; do
                    log "  - $obj"
                done
            fi

            # Ask for confirmation before cleanup
            read -p "Do you want to proceed with cleanup and delete all resources?
 (y/n): " confirm
            if [[ $confirm != [yY] && $confirm != [yY][eE][sS] ]]; then
                log "Cleanup aborted by user."
                return
            fi

            # Delete all versions of objects
            delete_all_versions "$BUCKET_NAME"

            # Delete the bucket
            log "Deleting bucket $BUCKET_NAME..."
            aws s3api delete-bucket --bucket "$BUCKET_NAME" || log "Warning: Failed
 to delete bucket"
        else
            log "Bucket $BUCKET_NAME does not exist, skipping cleanup"
        fi
    fi

    # Clean up local files
    log "Removing local files..."
    rm -f sample-file.txt sample-document.txt downloaded-sample-file.txt empty-
file.tmp

    log "Cleanup completed."
}

# Generate a random bucket name
generate_bucket_name() {
    local hex_id
    hex_id=$(openssl rand -hex 6)
    echo "demo-s3-bucket-$hex_id"
}
```

```
# Main script execution
main() {
    log "Starting Amazon S3 Getting Started Tutorial"

    # Generate a unique bucket name
    BUCKET_NAME=$(generate_bucket_name)
    log "Generated bucket name: $BUCKET_NAME"

    # Step 1: Create a bucket
    log "Step 1: Creating S3 bucket..."

    # Get the current region or default to us-east-1
    REGION=$(aws configure get region)
    REGION=${REGION:-us-east-1}
    log "Using region: $REGION"

    if [ "$REGION" = "us-east-1" ]; then
        aws s3api create-bucket --bucket "$BUCKET_NAME" || handle_error "Failed to
 create bucket"
    else
        aws s3api create-bucket \
            --bucket "$BUCKET_NAME" \
            --region "$REGION" \
            --create-bucket-configuration LocationConstraint="$REGION" ||
 handle_error "Failed to create bucket"
    fi
    log "Bucket created successfully"

    # Configure bucket settings
    log "Configuring bucket settings..."

    # Block public access (security best practice)
    log "Blocking public access..."
    aws s3api put-public-access-block \
        --bucket "$BUCKET_NAME" \
        --public-access-block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=tr
 || handle_error "Failed to configure public access block"

    # Enable versioning
    log "Enabling versioning..."
    aws s3api put-bucket-versioning \
        --bucket "$BUCKET_NAME" \
```

```
        --versioning-configuration Status=Enabled || handle_error "Failed to enable
versioning"

    # Set default encryption
    log "Setting default encryption..."
    aws s3api put-bucket-encryption \
        --bucket "$BUCKET_NAME" \
        --server-side-encryption-configuration '{"Rules":
[{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}' ||
handle_error "Failed to set encryption"

    # Step 2: Upload an object
    log "Step 2: Uploading objects to bucket..."

    # Create a sample file
    echo "This is a sample file for the S3 tutorial." > sample-file.txt

    # Upload the file
    aws s3api put-object \
        --bucket "$BUCKET_NAME" \
        --key "sample-file.txt" \
        --body "sample-file.txt" || handle_error "Failed to upload object"
    log "Object uploaded successfully"

    # Upload with metadata
    echo "This is a document with metadata." > sample-document.txt
    aws s3api put-object \
        --bucket "$BUCKET_NAME" \
        --key "documents/sample-document.txt" \
        --body "sample-document.txt" \
        --content-type "text/plain" \
        --metadata "author=AWSDocumentation,purpose=tutorial" || handle_error
"Failed to upload object with metadata"
    log "Object with metadata uploaded successfully"

    # Step 3: Download an object
    log "Step 3: Downloading object from bucket..."
    aws s3api get-object \
        --bucket "$BUCKET_NAME" \
        --key "sample-file.txt" \
        "downloaded-sample-file.txt" || handle_error "Failed to download object"
    log "Object downloaded successfully"

    # Check if an object exists
```

```
    log "Checking if object exists..."
    aws s3api head-object \
        --bucket "$BUCKET_NAME" \
        --key "sample-file.txt" || handle_error "Object does not exist"
    log "Object exists"

    # Step 4: Copy object to a folder
    log "Step 4: Copying object to a folder..."

    # Create a folder structure using a temporary empty file
    log "Creating folder structure..."
    touch empty-file.tmp
    aws s3api put-object \
        --bucket "$BUCKET_NAME" \
        --key "favorite-files/" \
        --body empty-file.tmp || handle_error "Failed to create folder"

    # Copy the object
    log "Copying object..."
    aws s3api copy-object \
        --bucket "$BUCKET_NAME" \
        --copy-source "$BUCKET_NAME/sample-file.txt" \
        --key "favorite-files/sample-file.txt" || handle_error "Failed to copy
object"
    log "Object copied successfully"

    # List objects in the bucket
    log "Listing all objects in the bucket..."
    aws s3api list-objects-v2 \
        --bucket "$BUCKET_NAME" \
        --query 'Contents[].Key' \
        --output table || handle_error "Failed to list objects"

    # List objects with a specific prefix
    log "Listing objects in the favorite-files folder..."
    aws s3api list-objects-v2 \
        --bucket "$BUCKET_NAME" \
        --prefix "favorite-files/" \
        --query 'Contents[].Key' \
        --output table || handle_error "Failed to list objects with prefix"

    # Add tags to the bucket
    log "Adding tags to the bucket..."
    aws s3api put-bucket-tagging \
```

```
        --bucket "$BUCKET_NAME" \
        --tagging 'TagSet=[{Key=Project,Value=S3Tutorial},
{Key=Environment,Value=Demo}]' || handle_error "Failed to add tags"
    log "Tags added successfully"

    log "Tutorial completed successfully!"
}

# Execute the main function
main
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - HeadObject
  - ListObjectVersions
  - ListObjectsV2
  - PutBucketEncryption
  - PutBucketTagging
  - PutBucketVersioning
  - PutObject
  - PutPublicAccessBlock

AWS STS exemples d'utilisation AWS CLI avec le script Bash

Les exemples de code suivants vous montrent comment effectuer des actions et implémenter des scénarios courants à l'aide du script AWS Command Line Interface with Bash with AWS STS.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code en contexte.

Rubriques

- Actions

Actions

`AssumeRole`

L'exemple de code suivant montre comment utiliser`AssumeRole`.

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
#  credentials.
#
# Parameters:
#       -n role_session_name -- The name of the session.
#       -r role_arn -- The ARN of the role to assume.
#
# Returns:
#       [access_key_id, secret_access_key, session_token]
#     And:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function sts_assume_role() {
  local role_session_name role_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary credentials:"
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
  }

  while getopts n:r:h option; do
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Pour plus de détails sur l'API, reportez-vous AssumeRoleà la section Référence des AWS CLI commandes.

Sécurité dans le AWS CLI

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Command Line Interface, voir AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS Command Line Interface (AWS CLI). Les rubriques suivantes expliquent comment configurer le pour répondre AWS CLI à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser le AWS CLI pour vous aider à surveiller et à sécuriser vos AWS ressources.

Rubriques

- Protection des données dans le AWS CLI
- Gestion de l'identité et des accès
- Validation de conformité pour ce AWS produit ou service
- Résilience pour ce AWS produit ou service
- Sécurité de l'infrastructure pour ce AWS produit ou service
- Appliquer une version minimale du protocole TLS pour AWS CLI

Protection des données dans le AWS CLI

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS Command Line Interface. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec AWS CLI ou d'autres Services AWS

utilisateurs de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement des données

Une caractéristique clé de tout service sécurisé est que les informations sont chiffrées lorsqu'elles ne sont pas utilisées activement.

Chiffrement au repos

Il AWS CLI ne stocke lui-même aucune donnée client autre que les informations d'identification dont il a besoin pour interagir avec les AWS services au nom de l'utilisateur.

Si vous utilisez le AWS CLI pour appeler un AWS service qui transmet les données clients à votre ordinateur local à des fins de stockage, reportez-vous au chapitre Sécurité et conformité du guide de l'utilisateur de ce service pour obtenir des informations sur la manière dont ces données sont stockées, protégées et cryptées.

Chiffrement en transit

Par défaut, toutes les données transmises depuis l'ordinateur client exécutant les points de terminaison du AWS service AWS CLI et sont cryptées en envoyant le tout via une connexion HTTPS/TLS.

Vous n'avez pas besoin de faire quoi que ce soit pour activer l'utilisation de HTTP/TLS. Il est toujours activé sauf si vous le désactivez explicitement pour une commande individuelle à l'aide de l'option de ligne de commande `--no-verify-ssl`.

Gestion de l'identité et des accès

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser AWS les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Comment Services AWS travailler avec IAM
- Résolution des problèmes AWS d'identité et d'accès

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez. AWS

Utilisateur du service : si vous avez l' Services AWS habitude de faire votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles AWS fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans AWS, consultez Résolution des problèmes AWS d'identité et d'accès le guide de l'utilisateur du Service AWS que vous utilisez.

Administrateur du service — Si vous êtes responsable des AWS ressources de votre entreprise, vous avez probablement un accès complet à AWS. C'est à vous de déterminer les AWS fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d’IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec AWS, consultez le guide de l'utilisateur Service AWS que vous utilisez.

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaiterez peut-être en savoir plus sur la façon d’écrire des politiques pour gérer l’accès à AWS. Pour consulter des exemples de politiques AWS basées sur l'identité que vous pouvez utiliser dans IAM, consultez le guide de l'utilisateur Service AWS que vous utilisez.

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez AWS Signature Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour plus d'informations, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Authentification multifactorielle AWS dans IAM dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez Tâches nécessitant des informations d'identification d'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des mots de passe et des clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons d'effectuer une rotation des clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin.

Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un rôle IAM (console). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service

peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.

- Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.
- Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal

(utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus

d’informations sur les limites d’autorisations, consultez Limites d’autorisations pour des entités IAM dans le Guide de l’utilisateur IAM.

- Politiques de contrôle des services (SCPs) : SCPs politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les Organizations et consultez SCPs les politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.
- Politiques de contrôle des ressources (RCPs) : RCPs politiques JSON que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les politiques IAM associées à chaque ressource que vous possédez. Le RCP limite les autorisations pour les ressources des comptes membres et peut avoir un impact sur les autorisations effectives pour les identités, y compris Utilisateur racine d'un compte AWS, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les Organizations RCPs, y compris une liste de ces Services AWS supports RCPs, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l’identité de l’utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d’une politique basée sur les ressources. Un refus explicite dans l’une de ces politiques annule l’autorisation. Pour plus d’informations, consultez Politiques de session dans le Guide de l’utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s’appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment Services AWS travailler avec IAM

Pour obtenir une vue d'ensemble du Services AWS fonctionnement de la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Pour savoir comment utiliser un service spécifique Service AWS avec IAM, consultez la section sécurité du guide de l'utilisateur du service concerné.

Résolution des problèmes AWS d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS ressources

Je ne suis pas autorisé à effectuer une action dans AWS

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `awes:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `awes:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS en charge, consultezComment Services AWS travailler avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.

- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Validation de conformité pour ce AWS produit ou service

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Conformité et gouvernance de la sécurité : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- Référence des services éligibles HIPAA : liste les services éligibles HIPAA. Tous ne Services AWS sont pas éligibles à la loi HIPAA.
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans plusieurs cadres (notamment le National Institute of Standards and Technology (NIST), le Payment

Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).

- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Ce AWS produit ou service suit le modèle de responsabilité partagée par le biais des services Amazon Web Services (AWS) spécifiques qu'il prend en charge. Pour obtenir des informations sur la sécurité des AWS services, consultez la AWS page de documentation sur la sécuritéAWS des services et les services concernés par les efforts de AWS conformité par programme de conformité.

Résilience pour ce AWS produit ou service

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité.

Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant.

Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d’une zone à l’autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l’échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Ce AWS produit ou service suit le modèle de responsabilité partagée par le biais des services Amazon Web Services (AWS) spécifiques qu'il prend en charge. Pour obtenir des informations sur la sécurité des AWS services, consultez la AWS page de documentation sur la sécuritéAWS des services et les services concernés par les efforts de AWS conformité par programme de conformité.

Sécurité de l'infrastructure pour ce AWS produit ou service

Ce AWS produit ou service utilise des services gérés et est donc protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à ce AWS produit ou service via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Ce AWS produit ou service suit le modèle de responsabilité partagée par le biais des services Amazon Web Services (AWS) spécifiques qu'il prend en charge. Pour obtenir des informations sur la sécurité des AWS services, consultez la AWS page de documentation sur la sécuritéAWS des services et les services concernés par les efforts de AWS conformité par programme de conformité.

Appliquer une version minimale du protocole TLS pour AWS CLI

Lorsque vous utilisez le AWS Command Line Interface (AWS CLI), le protocole Transport Layer Security (TLS) joue un rôle crucial dans la sécurisation des communications entre AWS CLI et Services AWS. Pour renforcer la sécurité lors de la communication avec les AWS services, vous devez utiliser le protocole TLS 1.2 ou version ultérieure.

AWS CLI la version 2 utilise un script Python interne qui est compilé pour utiliser au moins TLS 1.2 lorsque le service auquel il parle le prend en charge. Tant que vous utilisez la version 2 du AWS CLI, aucune autre étape n'est nécessaire pour appliquer ce minimum. Pour garantir une sécurité accrue, veillez à effectuer une mise à jour vers une version récente du AWS CLI.

Le AWS CLI and Service AWS peut échanger des données en toute sécurité, le protocole TLS assurant le cryptage, l'authentification et l'intégrité des données. En tirant parti du protocole TLS, vous AWS CLI garantissez que vos interactions Services AWS sont protégées contre les accès non autorisés et les violations de données, améliorant ainsi la sécurité globale de votre AWS écosystème.

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS Command Line Interface. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les Services AWS. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable de la configuration de la sécurité et des tâches de gestion de Services AWS ce que vous utilisez. Pour plus d'informations sur la protection des données, consultezthe section called “Protection des données”.

Guide de migration pour la AWS CLI version 2

Cette section contient des instructions pour mettre à jour la AWS CLI version 1 vers AWS CLI la version 2. La AWS CLI version 2 s'appuie sur la AWS CLI version 1 et inclut des fonctionnalités et des améliorations basées sur les commentaires de la communauté. La AWS CLI version 2 est la version majeure la plus récente AWS CLI et prend en charge toutes les dernières fonctionnalités. Certaines fonctionnalités introduites dans la version 2 ne sont pas rétroportées vers la version 1 et vous devez effectuer une mise à niveau pour accéder à ces fonctionnalités.

Pour éviter des problèmes inattendus, avant de migrer vers la version 2, renseignez-vous sur les différences entre les versions. La AWS CLI version 2 inclut de nouvelles fonctionnalités et des modifications qui peuvent vous obliger à mettre à jour vos scripts ou commandes pour des raisons de rétrocompatibilité.

AWS CLI les versions 1 et 2 utilisent le même nom de `aws` commande. Si les deux versions sont installées, votre ordinateur utilise la première qui se trouve sur votre chemin de recherche. Cela peut entraîner l'appel de votre ancienne AWS CLI version par le nom de votre `aws` commande, même si la nouvelle est installée.

Pour passer à la AWS CLI version 2, suivez l'une des instructions ci-dessous :

- Si vous avez déjà installé AWS CLI la version 1, suivez les instructions figurant dansthe section called “Instructions de migration”.
- Si vous n'avez pas encore installé AWS CLI la version 1, suivez les instructions figurant dansMise en route.

Rubriques

- Nouvelles fonctionnalités et modifications de la AWS CLI version 2
- Installation de AWS CLI la version 2 à partir de AWS CLI la version 1

Nouvelles fonctionnalités et modifications de la AWS CLI version 2

Cette rubrique décrit les nouvelles fonctionnalités et les changements de comportement entre les AWS CLI versions 1 et AWS CLI 2. Ces modifications peuvent vous obliger à mettre à jour vos scripts ou commandes afin d'obtenir le même comportement dans la version 2 que dans la version 1.

Rubriques

- AWS CLI nouvelles fonctionnalités de la version 2
- Changements majeurs entre AWS CLI la version 1 et AWS CLI la version 2

AWS CLI nouvelles fonctionnalités de la version 2

La AWS CLI version 2 est la version majeure la plus récente AWS CLI et prend en charge toutes les dernières fonctionnalités. Certaines fonctionnalités introduites dans la version 2 ne sont pas rétroportées vers la version 1 et vous devez effectuer une mise à niveau pour accéder à ces fonctionnalités. Les principales fonctions sont notamment :

L'interpréteur Python n'est pas nécessaire

La AWS CLI version 2 n'a pas besoin d'une installation séparée de Python. Il inclut une version intégrée.

Magiciens

Vous pouvez utiliser un assistant avec la AWS CLI version 2. L'assistant vous guide dans la construction de certaines commandes.

Authentification IAM Identity Center

Si votre organisation utilise AWS IAM Identity Center (IAM Identity Center), vos utilisateurs peuvent se connecter à Active Directory, à un annuaire IAM Identity Center intégré ou à un autre IdP connecté à IAM Identity Center. Ils sont ensuite mappés à un rôle AWS Identity and Access Management (IAM) qui vous permet d'exécuter AWS CLI des commandes.

Invite automatique

Lorsqu'elle est activée, la AWS CLI version 2 peut vous demander des commandes, des paramètres et des ressources lorsque vous exécutez une `aws` commande.

Exécution des images officielles Amazon ECR Public ou Docker pour AWS CLI

L'image Docker officielle du AWS CLI fournit une isolation, une portabilité et une sécurité qui sont AWS directement prises en charge et maintenues. Ainsi, vous pouvez utiliser la AWS CLI version 2 dans un environnement basé sur des conteneurs sans avoir à gérer vous-même l'installation.

Téléavertisseur côté client

La AWS CLI version 2 prévoit l'utilisation d'un programme de téléavertisseur côté client pour la sortie. Par défaut, cette fonctionnalité est activée et renvoie toutes les sorties via le programme de téléavertisseur par défaut de votre système d'exploitation.

`aws configure import`

Importer `.csv` les informations d'identification générées à partir du AWS Management Console. Un `.csv` fichier est importé avec le nom du profil correspondant au nom d'utilisateur IAM.

`aws configure list-profiles`

Répertorie les noms de tous les profils que vous avez configurés.

the section called “Format de sortie de flux YAML”

Le `yaml-stream` format `yaml` and tire parti du format YAML tout en offrant une visualisation plus réactive de grands ensembles de données en vous transmettant les données en streaming. Vous pouvez commencer à visualiser et à utiliser les données YAML avant le téléchargement complet de la requête.

Nouvelles **ddb** commandes de haut niveau pour DynamoDB

La AWS CLI version 2 contient les commandes Amazon `ddb put`DynamoDB de haut niveau et. `ddb select` Ces commandes fournissent une interface simplifiée pour placer des éléments dans des tables DynamoDB et effectuer des recherches dans une table ou un index DynamoDB.

`aws logs tail`

La AWS CLI version 2 dispose d'une `aws logs tail` commande personnalisée qui suit les journaux d'un groupe Amazon CloudWatch Logs. Par défaut, la commande renvoie les journaux de tous les flux de CloudWatch journaux associés au cours des dix dernières minutes.

Ajout du support des métadonnées pour les **s3** commandes de haut niveau

La AWS CLI version 2 ajoute le `--copy-props` paramètre aux `s3` commandes de haut niveau. Ce paramètre vous permet de configurer des métadonnées et des balises supplémentaires pour Amazon Simple Storage Service (Amazon S3).

`AWS_REGION`

La AWS CLI version 2 possède une variable d'environnement AWS compatible avec le SDK appelée. `AWS_REGION` Cette variable indique Région AWS à qui envoyer les demandes. Elle remplace la variable d'`AWS_DEFAULT_REGION`environnement, qui n'est applicable que dans le AWS CLI.

Changements majeurs entre AWS CLI la version 1 et AWS CLI la version 2

Cette section décrit tous les changements de comportement entre la AWS CLI version 1 et AWS CLI la version 2. Ces modifications peuvent vous obliger à mettre à jour vos scripts ou commandes afin d'obtenir le même comportement dans la version 2 que dans la version 1.

Rubriques

- Variable d'environnement ajoutée pour définir le codage du fichier texte
- Les paramètres binaires sont transmis sous forme de chaînes codées en base64 par défaut
- Amélioration de la gestion par Amazon S3 des propriétés des fichiers et des balises pour les copies partitionnées
- Aucune récupération automatique de http:// ou https:// URLs pour les paramètres
- Pager utilisé pour toutes les sorties par défaut
- Les valeurs de sortie de l'horodatage sont normalisées au format ISO 8601
- Gestion améliorée des CloudFormation déploiements qui n'entraînent aucune modification
- Comportement par défaut modifié pour le point de terminaison Amazon S3 régional pour us-east-1 la région
- Comportement par défaut modifié pour les points de AWS STS terminaison régionaux
- ecr get-loginretiré et remplacé par ecr get-login-password
- AWS CLI le support de la version 2 pour les plugins est en train de changer
- Suppression du support des alias masqués
- Le paramètre du fichier api_versions de configuration n'est pas pris en charge
- AWS CLI la version 2 utilise uniquement Signature v4 pour authentifier les demandes Amazon S3
- AWS CLI la version 2 est plus cohérente avec les paramètres de pagination
- AWS CLI la version 2 fournit des codes de retour plus cohérents pour toutes les commandes

Variable d'environnement ajoutée pour définir le codage du fichier texte

Par défaut, les fichiers texte the section called “BLOB” utilisent le même encodage que les paramètres régionaux installés. Comme la AWS CLI version 2 utilise une version intégrée de Python, les variables d'`PYTHONIOENCODING`environnement `PYTHONUTF8` et ne sont pas prises en charge. Pour définir un encodage de fichier texte différent de celui des paramètres régionaux, utilisez la

variable d'environnement `AWS_CLI_FILE_ENCODING`. L'exemple suivant montre comment AWS CLI ouvrir des fichiers texte `UTF-8` sous Windows.

```
AWS_CLI_FILE_ENCODING=UTF-8
```

Pour plus d'informations, consultez Configuration des variables d'environnement pour AWS CLI.

Les paramètres binaires sont transmis sous forme de chaînes codées en base64 par défaut

Dans le AWS CLI, certaines commandes nécessitaient des chaînes codées en base64, tandis que d'autres nécessitaient des chaînes d'octets codées en UTF-8. Dans la AWS CLI version 1, le transfert de données entre deux types de chaînes codées nécessitait souvent un traitement intermédiaire. La AWS CLI version 2 rend la gestion des paramètres binaires plus cohérente, ce qui permet de transmettre les valeurs d'une commande à l'autre de manière plus fiable.

Par défaut, la AWS CLI version 2 transmet tous les paramètres binaires d'entrée et de sortie binaires sous forme de chaîne codée en base64 `blobs` (gros objet binaire). Pour de plus amples informations, veuillez consulter the section called "BLOB".

Pour revenir au comportement de la AWS CLI version 1, utilisez la configuration du `cli_binary_format` fichier ou le `--cli-binary-format` paramètre.

Amélioration de la gestion par Amazon S3 des propriétés des fichiers et des balises pour les copies partitionnées

Lorsque vous utilisez les commandes de la AWS CLI version 1 dans l'`aws s3`espace de noms pour copier un fichier d'un emplacement de compartiment S3 vers un autre, et que cette opération utilise une copie en plusieurs parties, aucune propriété de fichier de l'objet source n'est copiée vers l'objet de destination.

Par défaut, les commandes correspondantes de la AWS CLI version 2 transfèrent toutes les balises et certaines propriétés de la copie source vers la copie de destination. Par rapport à la AWS CLI version 1, cela peut entraîner un plus grand nombre d'appels d' AWS API vers le point de terminaison Amazon S3. Pour modifier le comportement par défaut des `s3` commandes de AWS CLI la version 2, utilisez le `--copy-props` paramètre.

Pour de plus amples informations, veuillez consulter the section called "Propriétés des fichiers et balises dans les copies en plusieurs parties".

Aucune récupération automatique de **`http://`** ou **`https://`** URLs pour les paramètres

La AWS CLI version 2 n'effectue aucune `GET` opération lorsqu'une valeur de paramètre commence par `http://` ou`https://`, et n'utilise pas le contenu renvoyé comme valeur de paramètre. Par conséquent, l'option de ligne de commande associée `cli_follow_urlparam` est supprimée de la AWS CLI version 2.

Si vous devez récupérer une URL et transmettre le contenu de l'URL dans une valeur de paramètre, nous vous recommandons `curl` d'utiliser un outil similaire pour télécharger le contenu de l'URL dans un fichier local. Utilisez ensuite la `file://` syntaxe pour lire le contenu de ce fichier et utilisez-le comme valeur de paramètre.

Par exemple, la commande suivante n'essaie plus de récupérer le contenu de la page se trouvant dans `http://www.example.com` et de le transmettre en tant que paramètre. Au lieu de cela, elle transmet la chaîne de texte littéral `https://example.com` en tant que paramètre.

```
$ aws ssm put-parameter \
    --value http://www.example.com \
    --name prod.microservice1.db.secret \
    --type String 2
```

Si vous devez récupérer et utiliser le contenu d'une URL Web en tant que paramètre, vous pouvez effectuer les opérations suivantes dans la version 2.

```
$ curl https://my.example.com/mypolicyfile.json -o mypolicyfile.json
$ aws iam put-role-policy \
    --policy-document file://./mypolicyfile.json \
    --role-name MyRole \
    --policy-name MyReadOnlyPolicy
```

Dans l'exemple précédent, le `-o` paramètre indique `curl` d'enregistrer le fichier dans le dossier actuel sous le même nom que le fichier source. La deuxième commande récupère le contenu de ce fichier téléchargé et le transmet en tant que valeur de `--policy-document`.

Pager utilisé pour toutes les sorties par défaut

Par défaut, la AWS CLI version 2 renvoie toutes les sorties via le programme de téléavertisseur par défaut de votre système d'exploitation. Ce programme est le `less`programme sous Linux ou

macOS, et le `more`programme sous Windows. Cela peut vous aider à parcourir une grande quantité de résultats d'un service en affichant ces résultats page par page.

Vous pouvez configurer la AWS CLI version 2 pour utiliser un programme de pagination différent ou aucun programme de pagination du tout. Pour de plus amples informations, veuillez consulter the section called “Téléavertisseur côté client”.

Les valeurs de sortie de l'horodatage sont normalisées au format ISO 8601

Par défaut, la AWS CLI version 2 renvoie toutes les valeurs de réponse d'horodatage au format ISO 8601. Dans AWS CLI la version 1, les commandes renvoyaient des valeurs d'horodatage quel que soit le format renvoyé par la réponse de l'API HTTP, qui pouvait varier d'un service à l'autre.

Pour voir les horodatages au format renvoyé par la réponse de l'API HTTP, utilisez la `wire` valeur de votre `config` fichier. Pour de plus amples informations, veuillez consulter `cli_timestamp_format`.

Gestion améliorée des CloudFormation déploiements qui n'entraînent aucune modification

Par défaut, dans la AWS CLI version 1, si vous déployez un AWS CloudFormation modèle qui n'entraîne aucune modification, un code d'erreur d'échec est renvoyé. AWS CLI Cela pose des problèmes si vous ne considérez pas qu'il s'agit d'une erreur et que vous souhaitez que votre script continue. Vous pouvez contourner ce problème dans la AWS CLI version 1 en ajoutant le drapeau`--no-fail-on-empty-changeset`, qui revient`0`.

Comme il s'agit d'un cas d'utilisation courant, la AWS CLI version 2 renvoie par défaut un code de sortie réussi `0` lorsqu'aucun changement n'est causé par un déploiement et que l'opération renvoie un ensemble de modifications vide.

Pour revenir au comportement d'origine, ajoutez le drapeau`--fail-on-empty-changeset`.

Comportement par défaut modifié pour le point de terminaison Amazon S3 régional pour **`us-east-1`** la région

Lorsque vous configurez la AWS CLI version 1 pour utiliser la `us-east-1` région, elle AWS CLI utilise le point de `s3.amazonaws.com` terminaison global qui est physiquement hébergé dans la `us-east-1` région. La AWS CLI version 2 utilise le véritable point de terminaison régional `s3.us-east-1.amazonaws.com` lorsque cette région est spécifiée. Pour forcer la AWS CLI version 2

à utiliser le point de terminaison global, vous pouvez définir la région d'une commande suraws-global.

Comportement par défaut modifié pour les points de AWS STS terminaison régionaux

Par défaut, la AWS CLI version 2 envoie toutes les AWS Security Token Service (AWS STS) demandes d'API au point de terminaison régional pour l'API actuellement configurée Région AWS.

Par défaut, la AWS CLI version 1 envoie des AWS STS demandes au point de AWS STS terminaison global. Vous pouvez contrôler ce comportement par défaut dans la version 1 à l'aide du sts_regional_endpointsparamètre.

ecr get-loginretiré et remplacé par **ecr get-login-password**

La AWS CLI version 2 remplace la commande `aws ecr get-login` par la `aws ecr get-login-password` commande qui améliore l'intégration automatisée avec l'authentification des conteneurs.

La commande `aws ecr get-login-password` réduit le risque d'exposer vos informations d'identification dans la liste des processus, l'historique du shell ou d'autres fichiers journaux. Il améliore également la compatibilité avec la `docker login` commande pour une meilleure automatisation.

La `aws ecr get-login-password` commande est disponible dans les AWS CLI versions 1.17.10 et ultérieures, ainsi que dans la AWS CLI version 2. La `aws ecr get-login` commande précédente est toujours disponible dans la AWS CLI version 1 pour des raisons de rétrocompatibilité.

Avec la `aws ecr get-login-password` commande, vous pouvez remplacer le code suivant qui permet de récupérer un mot de passe.

```
$ (aws ecr get-login --no-include-email)
```

Pour réduire le risque d'exposer le mot de passe à l'historique ou aux journaux du shell, utilisez plutôt l'exemple de commande suivant. Dans cet exemple, le mot de passe est transféré directement à la commande `docker login`, où il est affecté au paramètre password par l'option `--password-stdin`.

```
$ aws ecr get-login-password | docker login --username AWS --password-stdin MY-
REGISTRY-URL
```

Pour plus d'informations, consultez `aws ecr get-login-password`le guide de référence de la AWS CLI version 2.

AWS CLI le support de la version 2 pour les plugins est en train de changer

La prise en charge des plugins dans la AWS CLI version 2 est entièrement provisoire et vise à aider les utilisateurs à migrer depuis la AWS CLI version 1 jusqu'à ce qu'une interface de plugin stable et mise à jour soit publiée. Rien ne garantit qu'un plugin particulier ou même l'interface du AWS CLI plugin seront pris en charge dans les futures versions de la AWS CLI version 2. Si vous utilisez des plugins, assurez-vous de vous connecter à une version spécifique de votre plugin AWS CLI et de tester ses fonctionnalités lors de la mise à niveau.

Pour activer la prise en charge du plug-in, créez une section `[plugins]` dans votre `~/.aws/config`.

```
[plugins]
cli_legacy_plugin_path = <path-to-plugins>/python3.7/site-packages
<plugin-name> = <plugin-module>
```

Dans `[plugins]` cette section, définissez la `cli_legacy_plugin_path` variable et définissez sa valeur sur le chemin des packages du site Python où se trouve votre module de plugin. Vous pouvez ensuite configurer un plugin en fournissant un nom pour le plugin (`plugin-name`) et le nom de fichier du module Python (`plugin-module`) qui contient le code source de votre plugin. AWS CLI Charge chaque plugin en l'important `plugin-module` et en appelant sa `awscli_initialize` fonction.

Suppression du support des alias masqués

AWS CLI la version 2 ne prend plus en charge les alias cachés suivants qui étaient pris en charge dans la version 1.

Dans le tableau suivant, la première colonne affiche le service, la commande et le paramètre qui fonctionnent dans toutes les versions, y compris la AWS CLI version 2. La deuxième colonne affiche l'alias qui ne fonctionne plus dans la AWS CLI version 2.

Service, commande et paramètre de travail	Alias obsolète
cognito-identity -arns create-identity-pool open-id-connect-provider	open-id-connect-provider-ar-ns
storagegateway describe-tapes tape-arns	tape-ar-ns

Service, commande et paramètre de travail	Alias obsolète
passerelle de stockage. describe-tape-archives.ruban adhésif	tape-ar-ns
passerelle de stockage. describe-vtl-devices. vtl-device-arns	vtl-device-ar-ns
passerelle de stockage. describe-cached-iscsi-volumes.volume-arns	volume-ar-ns
passerelle de stockage. describe-stored-iscsi-volumes.volume-arns	volume-ar-ns
route53domains.view-billing.start-time	démarrer
déployer. create-deployment-groupensemble de balises .ec2	ec-2-tag-set
déployer. list-application-revisions.s3-seau	s-3-bucket
déployer. list-application-revisions.s3-key-préfixe	s-3-key-prefix
déployer. update-deployment-groupensemble de balises .ec2	ec-2-tag-set
iam. enable-mfa-device.code d'authentification1	authentication-code-1
iam. enable-mfa-device.code d'authentification 2	authentication-code-2
iam. resync-mfa-device.code d'authentification1	authentication-code-1
iam. resync-mfa-device.code d'authentification 2	authentication-code-2
import/export. get-shipping-label.rue 1	street-1
import/export. get-shipping-label.rue 2	street-2
import/export. get-shipping-label.rue 3	street-3
lambda.publish-version.code-sha256	code-sha-256
voile lumineuse. import-key-pair. public-key-base64	public-key-base-64
opsworks.register-volume.ec2-volume-id	ec-2-volume-id

Le paramètre du fichier **`api_versions`** de configuration n'est pas pris en charge

La AWS CLI version 2 ne prend pas en charge l'appel de versions antérieures du AWS service à APIs l'aide du paramètre du fichier de `api_versions` configuration. Toutes les AWS CLI commandes appellent désormais la dernière version du service APIs actuellement prise en charge par le point de terminaison.

AWS CLI la version 2 utilise uniquement Signature v4 pour authentifier les demandes Amazon S3

La AWS CLI version 2 ne prend pas en charge les algorithmes de signature antérieurs pour authentifier de manière cryptographique les demandes de service envoyées aux points de terminaison Amazon S3. Cette signature s'effectue automatiquement à chaque demande Amazon S3 et seul le processus de signature Signature version 4 est pris en charge. Vous ne pouvez pas configurer la version de signature. Tous les compartiments Amazon S3 présignés utilisent URLs désormais uniquement le protocole SigV4 et ont une durée d'expiration maximale d'une semaine.

AWS CLI la version 2 est plus cohérente avec les paramètres de pagination

Dans la AWS CLI version 1, si vous spécifiez les paramètres de pagination sur la ligne de commande, la pagination automatique est désactivée comme prévu. Toutefois, lorsque vous spécifiez des paramètres de pagination à l'aide d'un fichier contenant le `--cli-input-json` paramètre, la pagination automatique n'était pas désactivée, ce qui pouvait entraîner une sortie inattendue. La AWS CLI version 2 désactive la pagination automatique, quelle que soit la manière dont vous fournissez les paramètres.

AWS CLI la version 2 fournit des codes de retour plus cohérents pour toutes les commandes

La AWS CLI version 2 est plus cohérente dans toutes les commandes et renvoie correctement un code de sortie approprié par rapport à la AWS CLI version 1. Nous avons également ajouté les codes de sortie 252, 253 et 254. Pour plus d'informations sur les codes de sortie, consultezthe section called “Codes de retour”.

Si vous êtes tributaire de la façon dont la AWS CLI version 1 utilise les valeurs des codes de retour, nous vous recommandons de vérifier les codes de sortie pour vous assurer que vous obtenez les valeurs que vous attendez.

Installation de AWS CLI la version 2 à partir de AWS CLI la version 1

Cette rubrique fournit des instructions pour la migration de la AWS CLI version 1 vers la AWS CLI version 2.

AWS CLI les versions 1 et 2 utilisent le même nom de `aws` commande. Si les deux versions sont installées, votre ordinateur utilise la première qui se trouve sur votre chemin de recherche. Si vous avez déjà installé AWS CLI la version 1, nous vous recommandons d'effectuer l'une des opérations suivantes pour utiliser la AWS CLI version 2 :

- Recommandé — Désinstallez AWS CLI la version 1 et utilisez uniquement AWS CLI la version 2.
- Pour installer les deux versions, utilisez la capacité de votre système d'exploitation à créer un lien symbolique (lien symbolique) ou un alias portant un nom différent pour l'une des deux `aws` commandes.

Pour plus d'informations sur les modifications majeures entre la version 1 et la version 2, consultezthe section called “Nouvelles fonctionnalités et modifications”.

Remplacement de la version 1 par la version 2

Procédez comme suit pour remplacer AWS CLI la version 1 par AWS CLI la version 2.

Pour remplacer AWS CLI la version 1 par AWS CLI la version 2

1. Préparez tous les scripts existants dont vous disposez pour la migration en confirmant toute modification importante entre la version 1 et la version 2 dansthe section called “Nouvelles fonctionnalités et modifications”.
2. Désinstallez la AWS CLI version 1 en suivant les instructions de désinstallation de votre système d'exploitation dans Installation, mise à jour et désinstallation de la AWS CLI version 1.
3. Vérifiez que le AWS CLI est complètement désinstallé à l'aide de la commande suivante.

   ```
   $ aws --version
   ```

   Effectuez l'une des opérations suivantes en fonction de la sortie :

- Aucune version renvoyée : vous avez correctement désinstallé la AWS CLI version 1 et vous pouvez passer à l'étape suivante.
- Une version est renvoyée : vous avez toujours une installation de la AWS CLI version 1. Pour obtenir les étapes de dépannage, consultez the section called “La commande « `aws --version` » renvoie une version après la désinstallation du AWS CLI”. Effectuez les étapes de dépannage jusqu'à ce qu'aucune sortie de version ne soit reçue.

4. Installez la AWS CLI version 2 en suivant les instructions d'installation appropriées à votre système d'exploitation dansInstallation ou mise à jour vers la dernière version du AWS CLI.

ide-by-sideInstallation S

Pour installer les deux versions, utilisez la capacité de votre système d'exploitation à créer un lien symbolique (lien symbolique) ou un alias portant un nom différent pour l'une des deux `aws` commandes.

1. Installez la AWS CLI version 2 en suivant les instructions d'installation appropriées à votre système d'exploitation dansInstallation ou mise à jour vers la dernière version du AWS CLI.
2. Utilisez la capacité de votre système d'exploitation à créer un lien symbolique ou un alias portant un nom différent pour l'une des deux `aws` commandes, par exemple en utilisant *`aws2`* pour AWS CLI la version 2. Voici des exemples de liens symboliques pour la AWS CLI version 2. Remplacez-le *`PATH`* par votre emplacement d'installation.

   Linux and macOS

   Vous pouvez utiliser un lien symbolique ou un alias sous Linux et macOS.

   ```
   $ alias aws2='PATH'
   ```

   Windows command prompt

   `DOSKEY`sous Windows.

   ```
   C:\> doskey aws2=PATH
   ```

Désinstaller la AWS CLI version 2

Cette rubrique décrit comment désinstaller la AWS Command Line Interface version 2 (AWS CLI version 2).

AWS CLI instructions de désinstallation de la version 2 :

Linux

Pour désinstaller, suivez la même méthode que celle que vous avez utilisée pour installer le AWS CLI.

Command line installer

Pour désinstaller la AWS CLI version 2, exécutez les commandes suivantes.

1. Localisez le lien symbolique et les chemins d'installation.

   - Utilisez la commande `which` pour rechercher le lien symbolique. Cela vous indique le chemin que vous avez utilisé avec le paramètre `--bin-dir`.

     ```
     $ which aws
     /usr/local/bin/aws
     ```

   - Utilisez la commande `ls` pour rechercher le répertoire vers lequel pointe le lien symbolique. Cela vous indique le chemin que vous avez utilisé avec le paramètre `--install-dir`.

     ```
     $ ls -l /usr/local/bin/aws
     lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
     ```

2. Supprimez les deux liens symboliques dans le répertoire `--bin-dir`. Si votre utilisateur dispose d'une autorisation d'écriture sur ces annuaires, vous n'avez pas besoin de les utiliser`sudo`.

   ```
   $ sudo rm /usr/local/bin/aws
   $ sudo rm /usr/local/bin/aws_completer
   ```

3. Supprimez le répertoire `--install-dir`. Si votre utilisateur dispose d'une autorisation d'écriture sur ce répertoire, vous n'avez pas besoin de l'utiliser`sudo`.

```
$ sudo rm -rf /usr/local/aws-cli
```

4. (Facultatif) Supprimez le AWS SDK partagé et AWS CLI les informations de paramètres dans le `.aws` dossier.

> **Warning**
>
> Ces paramètres de configuration et d'identification sont partagés entre tous AWS SDKs les AWS CLI. Si vous supprimez ce dossier, il ne sera plus accessible aux AWS SDKs personnes encore présentes sur votre système.

L'emplacement par défaut du `.aws` dossier varie selon les plateformes. Par défaut, le dossier se trouve dans*~/.aws/*. Si votre utilisateur dispose d'une autorisation d'écriture sur ce répertoire, vous n'avez pas besoin de l'utiliser`sudo`.

```
$ sudo rm -rf ~/.aws/
```

Snap

Nous fournissons une version officielle AWS prise en charge AWS CLI du`snap`. Si vous avez `snap` déjà installé le AWS CLI, procédez comme suit :

1. Exécutez la `snap remove` commande suivante pour AWS CLI.

```
$ snap remove aws-cli --classic
```

Si vous `sudo` l'avez déjà installé AWS CLI, vous devez l'ajouter à la commande.

```
$ sudo snap remove aws-cli --classic
```

2. Vous devrez peut-être redémarrer votre fenêtre d'invite de commandes ou votre ordinateur pour supprimer tous les fichiers.
3. (Facultatif) Supprimez le AWS SDK partagé et AWS CLI les informations de paramètres dans le `.aws` dossier.

> **Warning**
>
> Ces paramètres de configuration et d'identification sont partagés entre tous AWS SDKs les AWS CLI. Si vous supprimez ce dossier, il ne sera plus accessible aux AWS SDKs personnes encore présentes sur votre système.

L'emplacement par défaut du `.aws` dossier varie selon les plateformes. Par défaut, le dossier se trouve dans*~/.aws/*. Si vous avez des autorisations d'écriture dans ce répertoire, vous n'avez pas besoin de l'utiliser`sudo`.

```
$ sudo rm -r ~/.aws/
```

macOS

Pour désinstaller la AWS CLI version 2, exécutez les commandes suivantes en remplaçant les chemins que vous avez utilisés pour l'installation. Les exemples de commandes utilisent les chemins d'installation par défaut.

1. Recherchez le dossier qui contient les liens symboliques vers le programme principal et le compléteur.

   ```
   $ which aws
   /usr/local/bin/aws
   ```

2. À l'aide de ces informations, exécutez la commande suivante pour rechercher le dossier d'installation vers lequel pointent les liens symboliques.

   ```
   $ ls -l /usr/local/bin/aws
   lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/
   aws-cli/aws
   ```

3. Supprimez les deux liens symboliques dans le premier dossier. Si votre utilisateur possède déjà une autorisation d'écriture sur ces dossiers, vous n'avez pas besoin de les utiliser`sudo`.

   ```
   $ sudo rm /usr/local/bin/aws
   $ sudo rm /usr/local/bin/aws_completer
   ```

4. Supprimez le dossier d'installation principal. Utilisez `sudo` pour accéder en écriture au dossier `/usr/local`.

```
$ sudo rm -rf /usr/local/aws-cli
```

5. (Facultatif) Supprimez le AWS SDK partagé et AWS CLI les informations de paramètres dans le `.aws` dossier.

> **Warning**
>
> Ces paramètres de configuration et d'identification sont partagés entre tous AWS SDKs les AWS CLI. Si vous supprimez ce dossier, il ne sera plus accessible aux AWS SDKs personnes encore présentes sur votre système.

L'emplacement par défaut du `.aws` dossier varie selon les plateformes. Par défaut, le dossier se trouve dans*~/.aws/*. Si votre utilisateur dispose d'une autorisation d'écriture sur ce répertoire, vous n'avez pas besoin de l'utiliser`sudo`.

```
$ sudo rm -rf ~/.aws/
```

Windows

1. Ouvrez Programmes et fonctionnalités en effectuant l'une des opérations suivantes :
   - Ouvrez le Panneau de configuration, puis choisissez Programmes et fonctionnalités.
   - Ouvrez une invite de commandes, puis entrez la commande suivante.

```
C:\> appwiz.cpl
```

2. Sélectionnez l'entrée nommée AWS Command Line Interface et cliquez sur Désinstaller pour lancer le programme de désinstallation.
3. Confirmez que vous souhaitez désinstaller le AWS CLI.
4. (Facultatif) Supprimez le AWS SDK partagé et AWS CLI les informations de paramètres dans le `.aws` dossier.

> **Warning**
>
> Ces paramètres de configuration et d'identification sont partagés entre tous AWS SDKs les AWS CLI. Si vous supprimez ce dossier, il ne sera plus accessible aux AWS SDKs personnes encore présentes sur votre système.

L'emplacement par défaut du `.aws` dossier varie selon les plateformes. Par défaut, le dossier se trouve dans*`%UserProfile%\.aws`*.

```
$ rmdir %UserProfile%\.aws
```

Résolution des erreurs AWS CLI d'installation et de désinstallation

Si vous rencontrez des problèmes après l'installation ou la désinstallation du AWS CLI, consultez les étapes de résolution the section called “Résoudre les erreurs” des problèmes. Pour connaître les étapes de résolution des problèmes les plus pertinentesthe section called “Erreurs de commande non trouvées”, reportez-vous aux the section called “La commande « `aws --version` » renvoie une version différente de celle que vous avez installée” sections, etthe section called “La commande « `aws --version` » renvoie une version après la désinstallation du AWS CLI”.

AWS CLI historique du document du guide de l'utilisateur

Le tableau suivant décrit les ajouts importants au guide de AWS Command Line Interface l'utilisateur à compter de janvier 2019. Pour recevoir les notifications sur les mises à jour de cette documentation, vous pouvez vous abonner au Flux RSS.

Modification	Description	Date
Support des terminaux basé sur les comptes ajouté à la version 2 AWS CLI	Les points de terminaison basés sur un compte utilisant votre identifiant de AWS compte sont désormais pris en charge dans la AWS CLI version 2.	20 mars 2025
Support des terminaux basé sur les comptes ajouté à la version 1 AWS CLI	Les points de terminaison basés sur un compte utilisant votre identifiant de AWS compte sont désormais pris en charge dans la AWS CLI version 2.	24 février 2025
Informations d'identification et d'authentification mises à jour.	Instructions et exemples de méthodes d'identification et d'authentification mis à jour. Cela inclut la mise à jour des pages de démarrage et des pages de configuration pertinentes. Pour tenir compte de cette augmentation de la documentation, les rubriques pertinentes relatives aux informations d'identification ont été déplacées vers la nouvelle section Authentification et informations d'accès .	31 mars 2023

Configuration du fournisseur de jetons avec actualisation automatique de l'authentification pour AWS IAM Identity Center ajout	Le nouveau processus de configuration du AWS CLI pour authentifier les utilisateurs AWS IAM Identity Center (IAM Identity Center) à l'aide de la configuration du fournisseur de jetons SSO, qui peut récupérer automatiquement les jetons d'authentification actualisés.	7 décembre 2022
Publication de l'image publique Amazon ECR officielle de la AWS CLI version 2	L'image publique Amazon ECR officiellement prise en charge pour la AWS CLI version 2 est publiée pour Linux, macOS et Windows.	18 novembre 2022
Mise à jour du guide de migration de la AWS CLI V1 vers la V2	Le guide des modifications les plus récentes a été étendu pour inclure des instructions de migration pour passer de la AWS CLI version 1 à la AWS CLI version 2. Inclut des mises à jour de la page de résolution des problèmes pour résoudre les problèmes d'installation.	13 mai 2022
Nouveau processus pour créer un AWS CLI programme d'installation à partir des sources.	Nouveau processus d'installation ou de mise à jour depuis la source vers la dernière version des AWS CLI systèmes d'exploitation pris en charge.	17 février 2022

Le contenu des versions AWS CLI V1 et V2 est désormais séparé dans leurs guides respectifs	Pour des raisons de clarté et de facilité, le contenu des AWS CLI AWS CLI versions 1 et 2 est désormais séparé dans leurs propres guides. Pour AWS CLI la version 1, consultez le guide de l'utilisateur de la AWS CLI version 1.	2 novembre 2021
Informations AWS CLI d'alias ajoutées	Informations AWS CLI d'alias ajoutées. Les alias sont des raccourcis que vous pouvez créer dans le AWS Command Line Interface (AWS CLI) pour raccourcir les commandes ou les scripts que vous utilisez fréquemment.	11 mars 2021
Informations de sortie du filtre mises à jour	Informations mises à jour pour les filtres et déplacées vers leur propre page.	1er février 2021
Informations ajoutées pour Wizards	Ajout d'informations sur l'assistant de AWS CLI version 2.	20 novembre 2020
Prompt automatique mis à jour	Mise à jour des informations de demande automatique de la AWS CLI version 2 avec les fonctionnalités actuelles.	10 novembre 2020
Ajout d'un exemple de script Amazon S3	Ajout d'un exemple de script de cycle de vie Amazon S3.	15 octobre 2020
Ajout d'un exemple de EC2 script Amazon	Ajout d'un exemple de script de type d' EC2 instance Amazon.	15 octobre 2020

Informations sur les nouvelles tentatives ajoutées	Ajout d'une page de nouvelles tentatives présentant les fonctionnalités et le comportement des tentatives dans le. AWS CLI	17 septembre 2020
Page de pagination côté serveur et côté client	Informations de pagination mises à jour et centralisées sur une seule page.	17 août 2020
Page de commandes s3 mise à jour	Mise à jour de la page des commandes s3 de haut niveau avec de nouveaux exemples et ressources.	30 juillet 2020
Informations d'installation mises à jour	Les informations d'installation, de mise à jour et de désinstal lation pour Linux, macOS et Windows ont été mises à jour.	19 mai 2020
Informations ajoutées pour le codage des fichiers texte sur la AWS CLI version 2	Par défaut, AWS CLI la version 2 utilise le même encodage de fichier texte que le fichier local. Vous pouvez désormais utiliser des variables d'environnement pour définir l'encodage des fichiers texte.	14 mai 2020
Image Docker officielle de la AWS CLI version 2 publiée	L'image Docker de support officielle pour la AWS CLI version 2 est publiée pour tous les systèmes Linux, macOS et Windows.	31 mars 2020

Ajout d'informations concernant les téléavertisseurs côté client pour la version 2 AWS CLI	Par défaut, AWS CLI la version 2 utilise le programme de téléavertisseur `less` pour toutes les sorties côté client.	19 février 2020
AWS Command Line Interface (AWS CLI) La version 2 est officiellement publiée	La AWS CLI version 2 est généralement disponible et est la version recommandée aux clients pour l'installation.	10 février 2020
Le programme d'installation de macOS pour AWS CLI la version 2 est désormais un `.pkg` fichier d'installation d'Apple Package.	Le programme d'installation de macOS pour AWS CLI la version 2 a été mis à jour à partir d'un `.zip` fichier contenant un script shell vers le package complet de macOS Installer. Cela simplifie l'installation et la rend compatible avec les dernières versions de macOS.	3 février 2020
Ajout de contenu pour améliorer la gestion par défaut des points de terminaison régionaux S3 et STS dans la AWS CLI version 2	Par défaut, AWS CLI la version 2 dirige désormais les demandes pour Amazon S3 et les AWS STS services vers le point de terminaison régional actuellement configuré plutôt que vers le point de terminaison global.	13 janvier 2020

Version préliminaire pour développeurs de AWS CLI la version 2	Annonce de la sortie en avant-première de AWS CLI la version 2. Ajout d'instructions sur l'installation de la version 2. Ajout d'une rubrique relative à la migration pour aborder les différences entre les versions 1 et 2.	7 novembre 2019
Ajout de la prise en charge AWS IAM Identity Center de AWS CLI deux profils nommés	AWS CLI la version 2 ajoute la prise en charge de la création d'un profil nommé qui peut se connecter directement à IAM Identity Center et obtenir des informations d'identification AWS temporaires à utiliser dans AWS CLI les commandes suivantes.	7 novembre 2019
Nouvelle section MFA	Une nouvelle section décrivant la manière d'accéder à l'interface de ligne de commande à l'aide de l'authentification multi-facteurs et des rôles a été ajoutée.	3 mai 2019
Mise à jour de la section « Utilisation de la CLI »	Des améliorations majeures et des ajouts ont été apportés aux instructions et procédures d'utilisation.	7 mars 2019
Mise à jour de la section « Installation de la CLI »	Améliorations et ajouts majeurs aux instructions et procédures AWS CLI d'installation.	7 mars 2019

Mise à jour de la section « Configuration de la CLI »	Améliorations et ajouts majeurs aux instructions et procédures de AWS CLI configuration.	7 mars 2019